Machine Learning and Optimization Techniques for Trapped-ion Quantum Simulators

by

Yi Hong Teoh

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Science
in
Physics

Waterloo, Ontario, Canada, 2020

© Yi Hong Teoh 2020

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of contributions

Yi Hong Teoh was the sole author of Chapters 1, 4 and 5. The content in these chapters were not written for publication. Chapters 2 and 3 consist in part of material written for publication.

For the research in Chapter 2:
This work was published in the journal, Quantum Science and Technology. This work was conducted under the supervision of Dr. Kazi Rajibul Islam and Dr. Roger Melko at the University of Waterloo. Marina Drygala contributed to the early stage development of the neural network used in this chapter. Yi Hong Teoh is the major contributor to the development of the neural network used in this chapter, the primary coder of the project and performed all analysis shown in this chapter. The initial draft of the manuscript was written by Yi Hong Teoh and further refined with the input of Dr. Kazi Rajibul Islam and Dr. Roger Melko.

Citation:
[1] Y. H. Teoh, M. Drygala, R. G. Melko, and R. Islam, Machine learning design of a trapped-ion quantum spin simulator, Quantum Science and Technology 5, 024001 (2020).

For the research in Chapter 3:
This work has been accepted for publication in the journal, Physical Review A. This work was conducted under the supervision of Dr. Kazi Rajibul Islam at the University of Waterloo. Dr. Manas Sajjan contributed in the development of an additional application of the proposal (not detailed in this thesis). Dr. Fereshteh Rajabi and Zewen Sun contributed some calculations used in the work. Yi Hong Teoh is the sole developer of the algorithm and code used in this chapter. All analysis shown in this chapter was performed by Yi Hong Teoh. Furthermore, Yi Hong Teoh is the primary author of the content detailed in this chapter with further revisions being done with the advice of Dr. Kazi Rajibul Islam, Dr. Fereshteh Rajabi, Dr. Manas Sajjan and Zewen Sun.

Citation:
[2] Y. H. Teoh, M. Sajjan, Z. Sun, F. Rajabi, and R. Islam, Manipulating phonons of a trapped-ion system using optical tweezers, Phys. Rev. A 104, 022420 (2021).

For the research in Chapter 4: This work was conducted under the supervision of Dr. Kazi Rajibul Islam at the University of Waterloo. Scott Hubele assisted in the development

of the algorithms and writing the code used in this chapter. Yi Hong Teoh is a major contributor in the development of the algorithm and code used in this chapter. Analyses shown in this chapter were performed by Yi Hong Teoh. Yi Hong Teoh is the sole author of the content in this chapter.

Abstract

In recent years, quantum simulators have been the focus of intense research due to their potential in unraveling the inner workings of complex quantum systems. The exponential scaling of the Hilbert space of quantum systems limits the capabilities of classical approaches. Quantum simulators, on the other hand, are suited to efficiently emulate these quantum systems for in-depth research. Out of the various platforms for quantum simulators, trapped-ions have been prized for their long coherence times, exceptional initialization and detection fidelities and their innate full-connectivity. In the development of the trapped-ion apparatus as a quantum simulator, numerous optimization problems emerge. In this thesis, we examine, develop and refine the strategies available to us for addressing such problems. A class of techniques of particular interest to us is the set of machine learning algorithms. This can be attributed to their capabilities at identifying correlations in massive data sets. Specifically, we describe how artificial neural networks can be employed to assist in the programming of the trapped-ion system as an arbitrary spin model quantum simulator. In addition, we present an augmentation to the trapped-ion architecture, utilizing optical tweezers, that enables the programmable manipulation of phonon modes. We describe the protocols developed to effectively program the phonon modes and propose an application of the scheme for enhancing the performance of multi-species systems. Finally, we explore the use of machine learning algorithms to perform state readout of the trapped-ion system. The work in this thesis extends the utility of the trapped-ion system for performing quantum information processing experiments.

Acknowledgements

First and foremost, I extend my sincerest gratitude to my supervisor Dr. Kazi Rajibul Islam for his patient guidance throughout my Masters program. Despite the fact that the COVID-19 pandemic has made the past year challenging, he has made an enormous effort to remain in contact and oversee my research. I am deeply appreciative of the research skills he has taught me.

Secondly, I convey my gratitude to Dr. Roger Melko, my co-supervisor, for assisting in the coordination of the collaborative effort between QITI and PIQuIL groups. His kind invitation to attend his group meetings has greatly broadened my horizons.

I would also like to express my appreciation to Dr. Ray Laflamme and Dr. Achim Kempf for taking the time to attend my committee meetings and review this thesis.

A huge thank you to all my colleagues in the QITI group, the feedback they have provided has been instrumental in conducting my research. Additionally, to the experimentalist of the group, Nikhil, Gilbert, Sainath, Roland, Nikolay and others, I am grateful for the knowledge they have imparted onto me regarding the experimental aspects of physics. To the theorist of the group, Zewen, Fereshteh, Manas, the discussions we had during our meetings have been incredibly informative and I have learnt so much from them.

I would like to thank the members of PIQuIL, Stefanie, Estelle, Roger, Ejaaz, Issac, Stewart. Their expertise in machine learning has proven invaluable in assisting me with my research.

Lastly, I am eternally grateful to my family for their unwavering support.

Dedication

This is dedicated to my family.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The advent of quantum theory in the 1900s has dramatically altered our understanding of the fundamental laws of the universe. In modern times, quantum theory has been instrumental in the advancement of technology. However, the study of quantum theory is intractable on classical systems, requiring an exponential amount of resources. To address this issue, Feynmann proposed a universal quantum simulator [3] which would be capable of emulating an arbitrary quantum system, with the appropriate programming, for study. Many platforms [4–7] have been proposed to be capable of such a feat, however, each system is accompanied by a unique set of challenges. In this thesis, we focus on the trapped-ion system which has proven itself as a leading platform for quantum infromation processing (QIP) experiments. This can be attributed to long coherence times [8], ease of qubit initialization and detection, and presence of long range interactions mediated by collective vibrational (phonon) modes. The latter property makes the trapped-ion system exceedingly flexible, capable of simulating numerous quantum systems of interest [9–12].

A disclaimer was previously mentioned regarding the requirement for the appropriate programming of the quantum simulator. In practice, the task of implementing such a programming is non-trivial and optimization of the experimental parameters is a prerequisite to accessing the versatility of the trapped-ion system. The emergence of optimization problems is not unique to the field of quantum information processing. Recently, machine learning has been at the forefront of efforts to tackle such optimization problems that naturally arise in the study of physics [13]. In these endeavours, machine learning has proven its capacity to effectively analyse data and identify the inherent patterns. As a result, machine learning is a valuable tool in the arsenal of a physicist.

1.1 Trapped-ion platform

In trapped-ion systems, we typically use an ion with a simple electronic structure. This encompasses the singly ionized alkali earth metal ions as they possess only a single valence electron making them hydrogen-like. We define the qubit states, $|0\rangle$ and $|1\rangle$, using the electronic states of the ion. There are two distinct paradigms for defining the qubit states. The first uses a stable ground state and a metastable excited state related by a transition with optical frequency, this is called an optical qubit. Whereas, the second employs two hyperfine ground states, this is called a hyperfine qubit. As an example, the usage of ytterbium ions, $^{171}\mathrm{Yb}^+$, is commonplace in the ion trapping community. The electronic structure of $^{171}\mathrm{Yb}^+$ is given in Fig. 1.1 and $^{171}\mathrm{Yb}^+$ forms a hyperfine qubit using the states $\left|S_{1/2}, F=0, m_F=0\right\rangle$ and $\left|S_{1/2}, F=1, m_F=0\right\rangle$. The transition between these two states are in the microwave spectrum. One can manipulate these states using microwave beams or two-photon Raman transition.

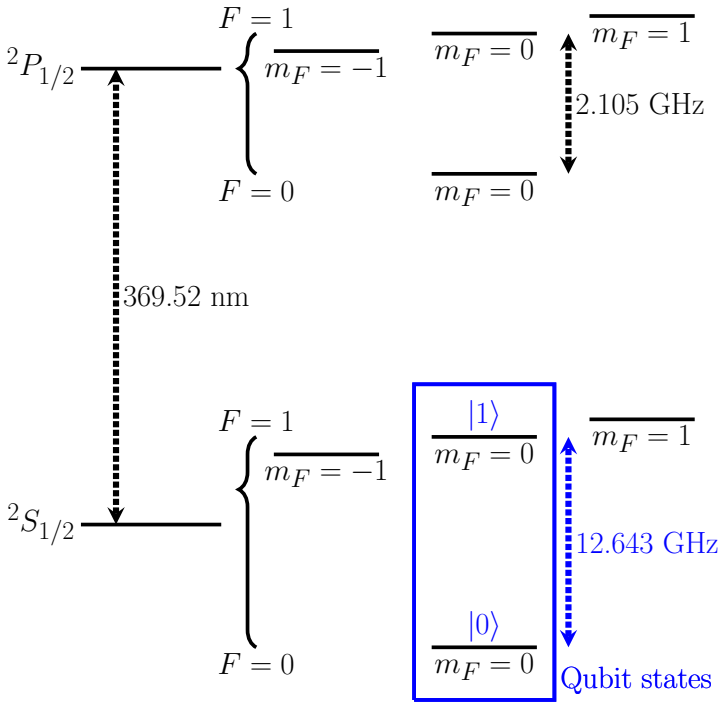


Figure 1.1: **Relevant energy levels of $^{171}\mathrm{Yb}^+$ for ion traps.** The hyperfine states, $\left|S_{1/2}, F=0, m_F=0\right\rangle$ and $\left|S_{1/2}, F=1, m_F=0\right\rangle$, are used to form the qubit states of the $^{171}\mathrm{Yb}^+$ ion.

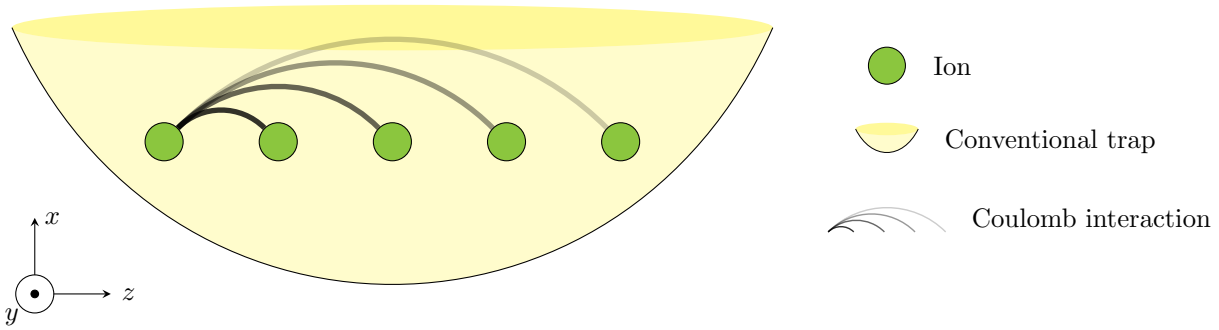


Figure 1.2: **Schematic of the potential in a conventional Paul trap.** Here, $N = 5$ ions are shown to be trapped by a harmonic pseudopotential generated by DC and RF electrodes, labelled in the diagram as the conventional trap. The ions also experience Coulomb interactions that couple the ions to one another, only the Coulomb coupling between the first ion and the other ions are shown, the remaining couplings have been omitted in the diagram for cleanliness.

Conventionally, a Paul (radio-frequency) trap is used to trap the ions using dynamic electric fields. In a Paul trap, direct current (DC) and radio-frequency (RF) electrodes are used to generate these dynamic electric fields. By taking the time average and harmonic approximation, the ions are confined by a time-averaged harmonic pseudopotential in the vicinity of the trap. In addition to the harmonic confinement, ions also experience mutual Coulomb repulsion. The total potential experienced by the ions in the Paul trap has the following form:

$$\phi^{\text{conv}} = \phi^{\text{Coulomb}} + \phi^{\text{DC+RF}}, \tag{1.1}$$

$$\phi^{\text{Coulomb}} = \sum_{i<j} \frac{q_i q_j}{4\pi\epsilon_0 \|\mathbf{r}_i - \mathbf{r}_j\|}, \tag{1.2}$$

$$\phi^{\text{DC+RF}} = \sum_i \frac{M_i}{2} \left(\omega_{x,i}^2 x_i^2 + \omega_{y,i}^2 y_i^2 + \omega_{z,i}^2 z_i^2\right). \tag{1.3}$$

Here, ϕ^{conv} is the total potential in the conventional trap, ϕ^{Coulomb} is the Coulomb potential between the ions, $\phi^{\text{DC+RF}}$ is the time-averaged harmonic pseudopotential generated by the DC and RF electrodes. The position, mass and charge of ion-i are represented by $\mathbf{r}_i$, M_i and q_i, respectively, and the vacuum permittivity by ϵ_0. A schematic of the trapped-ion

potential is shown in Fig. 1.1. The DC and RF induced trap frequencies on ion-i along the $\alpha \in \{x, y, z\}$ principal axes, $\omega_{\alpha,i}$, have the following form:

$$\omega_{z,i} = \sqrt{\frac{q_i V_{\mathrm{DC}} \xi_z}{M_i}}, \tag{1.4}$$

$$\omega_{x,i} = \sqrt{-\frac{q_i V_{\mathrm{DC}} \xi_x}{M_i} + \frac{q_i^2 V_{\mathrm{RF}}^2 \psi_x^2}{2 M_i^2 \Omega_{\mathrm{RF}}^2}}, \tag{1.5}$$

$$\omega_{y,i} = \sqrt{-\frac{q_i V_{\mathrm{DC}} \xi_y}{M_i} + \frac{q_i^2 V_{\mathrm{RF}}^2 \psi_y^2}{2 M_i^2 \Omega_{\mathrm{RF}}^2}}, \tag{1.6}$$

where ξ_α, ψ_α are the trap geometric factors, V_{DC} and V_{RF} are the DC and RF voltages and Ω_{RF} is the RF frequency. Notice that the axial (z) DC confinement engender a deconfinement in the radial (x, y) directions. This is a direct result of Earnshaw's theorem, an electrostatic field is incapable of trapping charged particles in all three dimensions.

Typically, QIP experiments are conducted in the regime where ions are trapped as a one-dimensional chain, without loss of generality, we allow this axis to be the axial (z) axis. We achieve the above configuration by setting the radial (x, y) trap frequencies to be considerably higher than the axial trap frequencies, i.e., $\omega_{x/y} \gg \omega_z$.

Earlier in the Introduction, we mentioned the flexibility the phonon modes in the trapped-ion system affords us. These phonon modes arise due to coulomb interactions between the ions. To observe the emergence of the phonon modes, the equation of motion for the potential given in Eq. 1.1 is solved for small perturbations about the equilibrium position. As the ions are displaced from their equilibrium configuration by an external field, they exhibit oscillatory behavior consistent with a linear combination of the phonon modes. The form of the phonon modes is obtained by studying the symmetrized mass-weighted Hessian of the potential evaluated at the equilibrium position of the ions, we dubbed this matrix the **A**-matrix. Under the assumption that the x, y and z axis form the principal axes of the system, the motion of the ions in the independent directions are uncoupled and the **A**-matrix can be decomposed into three sub-matrices, $\mathbf{A}^{(\alpha)}$ for $\alpha \in \{x, y, z\}$. The form of these **A**-matrices is as follows:

$$\mathrm{A}_{ij}^{(\alpha)} = \frac{1}{\sqrt{M_i}} \left[\mathrm{Hess}_\alpha \left(\phi^{\mathrm{conv}}\right)\right]_{ij}\Big|_{\{\mathbf{r}_i^*\}_i} \frac{1}{\sqrt{M_j}}. \tag{1.7}$$

Here, $[\mathrm{Hess}_\alpha(\,\cdot\,)]_{ij} = \partial^2 \cdot / \partial\alpha_i \partial\alpha_j$ denotes the Hessian operation for the α-coordinates.

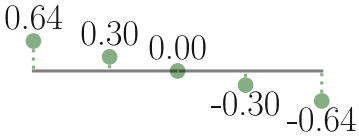
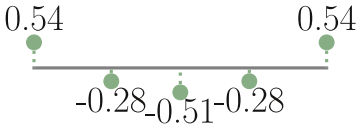
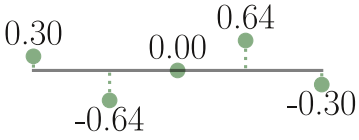
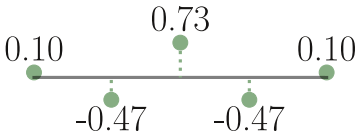
m	$\mathrm{B}_{im}^{(x/y)}$
1	
2	
3	
4	
5	

Table 1.1: **Typical transverse phonon mode eigenvectors.** The table shows the y-phonon modes possessed by a single species conventional trapped-ion system of $N = 5$ ions with trap frequencies, $\omega_x/2\pi = 1.2$ MHz, $\omega_y/2\pi = 1$ MHz and $\omega_z/2\pi = 0.2$ MHz. The numbers in the eigenvector plots correspond to normalized vectors that quantify the relative participation of each ion in the phonon modes.

By diagonalizing the $\mathbf{A}$-matrices, we obtain a set of eigenvalues and eigenvectors. The root of the eigenvalues, $\mathbf{w}^{(\alpha)}$, directly translate to the oscillation frequencies of the phonon modes. Additional consideration is required to determine the motional participation of each ion in the phonon modes. In the simple case of a trapped-ion system comprised of a single species, the eigenvectors, $\mathbf{B}^{(\alpha)}$, directly specify the motional participation of each ion in the phonon mode. Table 1.1 shows the typical transverse mode eigenvectors that emerge in this scenario. However, for a trapped-ion system composed of multiple species, an additional mass weighting has to be applied to obtain the motional participation of each ion in the phonon mode, derivation in Appendix A. Therefore, the motional participation

of an ion in the phonon modes depends on the mass-weighted eigenvector matrices with the following form:

$$\bar{\mathrm{B}}_{im}^{(\alpha)} = \frac{1}{\sqrt{M_i}} \mathrm{B}_{im}^{(\alpha)} \quad \text{normalized along index } i. \tag{1.8}$$

1.2 Machine Learning

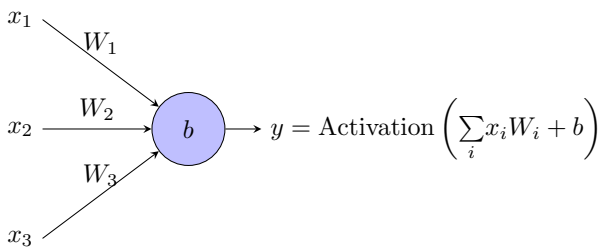


Figure 1.3: **Diagram of an artificial neuron.** An artificial neuron is a function that maps multiple inputs, x_i, to a single output, y, and consist of the following parameters: the weights, W_i, the bias, b, and the activation function, Activation($\cdot$).

Machine learning was developed with the idea that a computer is capable of learning the properties inherent in a data set and applying the knowledge learnt to make predictions. A subclass of machine learning algorithms is artificial neural networks. These neural networks took inspiration from the human brain, which is composed of neurons. In these neural networks, artificial neurons are the elementary building block. These artificial neurons are represented mathematically as functions with multiple inputs and a single output with form:

$$\text{Neuron}(x_i) = \text{Activation}\left(\sum_i W_i x_i + b\right). \tag{1.9}$$

Here, Activation($\cdot$) is the activation function for the neuron, W_i are the weights for the inputs, b_i is the bias of the neuron and x_i are the inputs to the neuron. Table 1.2 shows typical activation functions used by neurons, e.g. linear, rectified linear units (ReLU) and sigmoid.

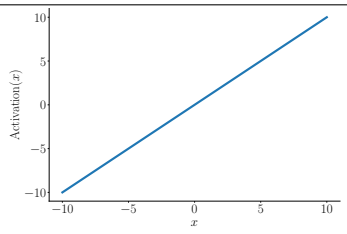
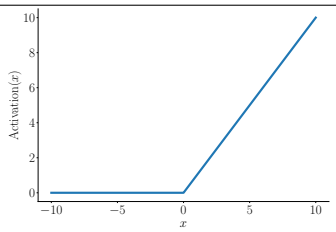
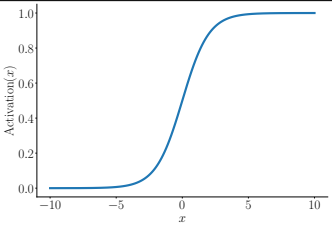
Activation function	Functional form	Graph
Linear	x	
ReLU	$\max(0, x)$	
Sigmoid	$\dfrac{1}{1+e^{-x}}$	

Table 1.2: **Typical activation functions for neurons.**

By grouping these neurons, we form the basis of a universal function approximator capable of reproducing any function given enough neurons. The structure of the grouping determines the architecture of the neural network.

Suppose the neurons are organized into layers, where each layer is fully connected with the preceding and following layers and information is only transmitted in one direction. Neural networks with such a structure are called feedforward neural networks (FFNN), diagram shown in Fig. 1.4.

FFNNs have been extensively used to perform classification. A FFNN is trained for this purpose in the following manner: A training data set, composed of data and the corresponding labels (supervised training), is first specified. The data is propagated through the neural network layers to obtain the outputs (predictions) of the neural network. A cost function is then defined to measure the performance of the neural network by comparing the labels and predictions. Finally, the weights and biases of the neural network are

adjusted by an algorithm, e.g. ADAM [14], ADADELTA [15], stochastic gradient descent (SGD), etc., to minimize the cost function. With the appropriate parameters, the neural network is capable of generalizing to data not encountered during training and make correct classifications.

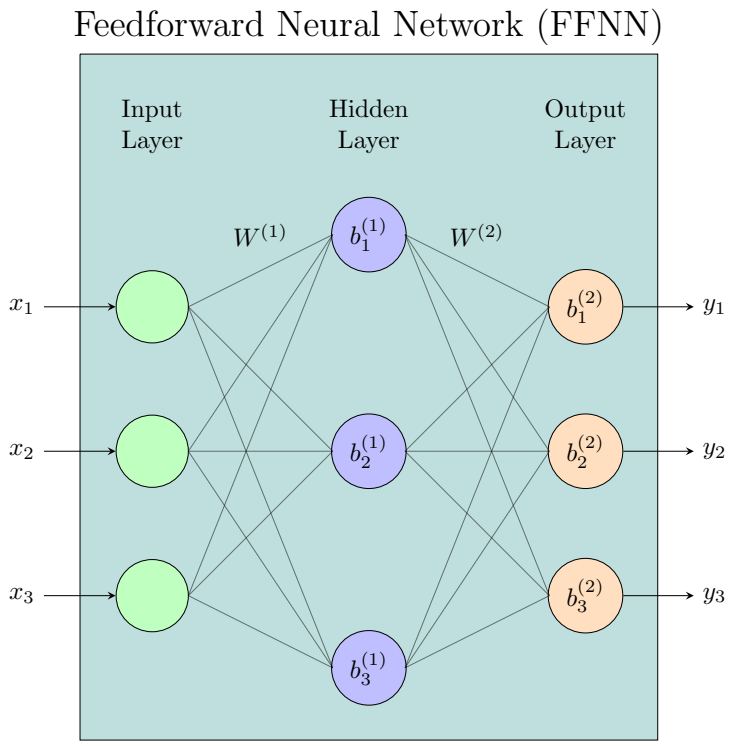


Figure 1.4: **Diagram of a feedforward neural network.** This neural network has an input, hidden and output layer, each consisting of 3 neurons. The parameters of the neural network include the weights between the input and hidden layer and between the hidden and output layer, $W^{(1)}$ and $W^{(2)}$, the biases of the hidden and output layer, $b^{(1)}$ and $b^{(2)}$. The hidden and output layer each has an associated activation function that is represented in the diagram by the color of the neurons.

Aside from FFNNs, there are convolutional neural networks (CNN), recurrent neural networks (RNN), restricted boltzmann machines (RBM), transformers, etc. Each of these neural network architectures have been tailored to address problems with different features. CNNs are commonly deployed in computer vision applications as they are highly effective at compressing high dimensional data into essential features, a requirement for

the efficient processing of image data. RNNs are frequently employed to perform language processing task as they are specialized at processing sequential data.

In addition to the various network architectures, there are also different paradigms for learning. We have encountered the first paradigm earlier, supervised learning. Here, the neural network is explicitly taught how to behave as labels are provided during the training phase. The second paradigm is unsupervised learning, where unstructured data is fed to the neural network. To identify the correlations present in the unstructured data, the neural network undergoes self-organization. Lastly, we have reinforcement learning, in which the neural network interacts with an environment and is rewarded for exhibiting desirable behaviors. Maximizing the cumulative reward is the objective of the neural network in this scenario.

1.3 Thesis outline

As mentioned previously, the development of the trapped-ion platform for quantum simulation experiments is accompanied by a multitude of optimization problems. In this thesis, we

- In Chapter 2: We will introduce the feedforward neural network developed for the purpose of programming a trapped-ion system for the simulation of an arbitrary spin-model. This neural network, after training, is able to achieve a high accuracy and produce results in a relatively short amount of time compared to traditional non-linear optimization routines.
- In Chapter 3: We will introduce a hybrid trapped-ion system that incorporates optical tweezers into the conventional trap detailed in Section 1.1. The resultant system has additional control over the phonon mode structure of the trapped-ion system. To program the phonon modes of such a system, we will introduce the novel algorithm we have developed. Additionally, we discuss an application of the scheme to enhance the coupling in a multi-species trapped-ion system.
- In Chapter 4: We will study the mechanism used by a feedforward neural network to perform state readout of a trapped-ion system. The feedforward neural network functions as a linear classifier and learns the point spread function (PSF) of the ions on the detector. Additionally, we introduce the algorithm we have developed to perform state readout that is capable of accommodating movements in the ions and varying number of ions.

Chapter 2

Machine learning programming a trapped-ion quantum spin simulator

Spin models are extremely versatile and the efficient simulation of spin models is highly sought after in multiple areas of physics. Spin models are currently under investigation for their computational potential in the field of optimization [16] and their ability to simulate problems in high energy physics [17]. Furthermore, higher dimensional spin models are used to model the crystallization of materials at low energy in condensed matter. Of particular interest are two-dimensional triangular or Kagome lattices, which exhibit a fascinating property of spin models, frustration. Therefore, the capability to simulate an arbitrary spin model greatly extends the usefulness of trapped-ion systems.

However, the simulation of an arbitrary spin model on a trapped-ion system requires many experimental parameters (laser frequencies and intensities) be tuned appropriately, this forms an inverse problem. Solving this inverse problem is non-trivial and in previous work [18], individual static spin models were solved for a system of N ions with non-linear optimization routines for the N^2 control parameters. However, such techniques are ineffective at simulating problems with dynamical interactions, such as quench and transport problems.

In recent years, machine learning has demonstrated its capacity to solve such inverse problems [19–21]. They do so by harnessing their power as a universal function approximator to approximate the form of the inverse function rather than optimizing for a specific target. This pseudo-inverse function can then be utilized to determine the parameters for other targets.

This chapter consist of work we published in the journal, Quantum Science and Technology

[1]. In this chapter, we developed a feedforward neural network (FFNN) that is capable of programming an arbitrary spin model onto a trapped-ion quantum simulator via the fine-tuning of the experimental parameters. We demonstrated that this simple but powerful neural network architecture, after training, is capable of identifying the control parameters to high accuracy, in a relatively short amount of time compared to previous non-linear optimization routines, for up to 50 ions on a consumer grade graphics processing unit (GPU) workstation.

2.1 Trapped-ions quantum spin simulator

In trapped-ion systems, the spin states are equated to the qubit states defined in Section 1.1. The Hamiltonian of the quantum spin-model we are interested in takes the form,

$$\hat{H} = \sum_{i<j} J_{ij} \hat{\sigma}_i^x \hat{\sigma}_j^x, \tag{2.1}$$

for an arbitrary interaction graph J_{ij}. For a system of N ions, there are $N(N-1)/2$ possible unique pairwise interactions J_{ij} in the Hamiltonian.

2.1.1 Mølmer-Sørensen Scheme

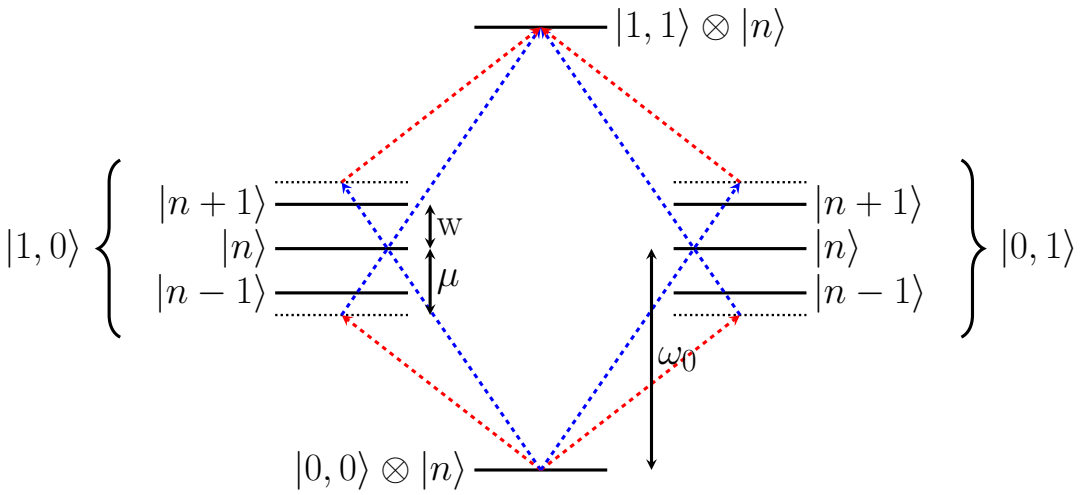


Figure 2.1: **Illustration of the Mølmer-Sørensen scheme.** Off-resonant blue and red detuned Raman beams, with detuning μ, couple the spin states and the phonon states. The net result is the coupling of the spin states of the ions without significant effect on the phonon states of the trapped-ion system.

The Mølmer-Sørensen scheme [22] is used to perform the simulation of a spin-model on a trapped-ion system. Symmetrically detuned Raman beams (blue and red sideband), of frequency $\omega_0 \pm \mu$, are used to couple the spin states of the ions to the phonon states, of frequency w, of a trapped-ion system off-resonantly. Here, ω_0 is the qubit frequency, μ is the Raman detuning. This process is illustrated in Fig. 2.1. As the Raman beams are detuned, the phonons are only virtually excited, i.e., $\ll 1$ phonon is excited by the scheme, and can be adiabatically eliminated [18, 23].

Instead of using direct beams to generate the Raman detunings, we consider individually addressing the ions with a bi-chromatic laser beam and applying a counter-propagating global beam. These beams interfere to generate a beat-note. We are able to tune the beat-note to the earlier mentioned frequency, $\omega_0 \pm \mu$, by adjusting the frequencies of the bi-chromatic laser beam and counter-propagating beam. The overall Hamiltonian of the trapped-ion system, in the presence of the optical field generated, would take the form of

Eq. 2.1, where the interaction graph, J_{ij}, would be given by,

$$J_{ij}(\Omega_i, \mu) = \Omega_i \Omega_j \sum_m \frac{\eta_{im}\eta_{jm}\mathrm{w}_m}{\mu^2 - \mathrm{w}_m^2}, \tag{2.2}$$

$$\eta_{im} = \sum_\alpha \mathrm{B}_{im}^{(\alpha)} \delta\mathrm{k}_\alpha \sqrt{\frac{\hbar}{2M\mathrm{w}_m}}. \tag{2.3}$$

Here, η_{im} is the Lamb-Dicke parameter, $\mathrm{B}_{im}^{(\alpha)}$ is the phonon mode eigenvector matrix in the α-direction, $\delta\mathbf{k}$ is the wave vector difference of the counter-propagating Raman lasers, w_m are the phonon mode frequencies, M is the mass of a single ion and Ω_i are the Rabi frequencies that characterize the strength of the interaction between the laser beam and each ion.

To generate an arbitrary spin-model in a system of N-ions, we generalize the scheme and apply N bi-chromatic laser beams onto each ion. These bi-chromatic beams interfere with the counter-propagating global laser beam to generate N Raman beat-notes. By setting each of the Raman beat-note detunings, μ_n, to be close to each of the N collective phonon modes, w_m, we induce coupling of the N spin states with the N phonon modes of the system. The form of the simulated interaction graph, J_{ij}, would be a sum of the interaction graph generated by each Raman beat-note detuning, i.e.,

$$J_{ij}(\Omega_{in}, \mu_n) = \sum_n^N \Omega_{in}\Omega_{jn} \sum_m^N \frac{\eta_{im}\eta_{jm}\mathrm{w}_m}{\mu_n^2 - \mathrm{w}_m^2}. \tag{2.4}$$

A diagram of this scheme is shown in Fig. 2.2.

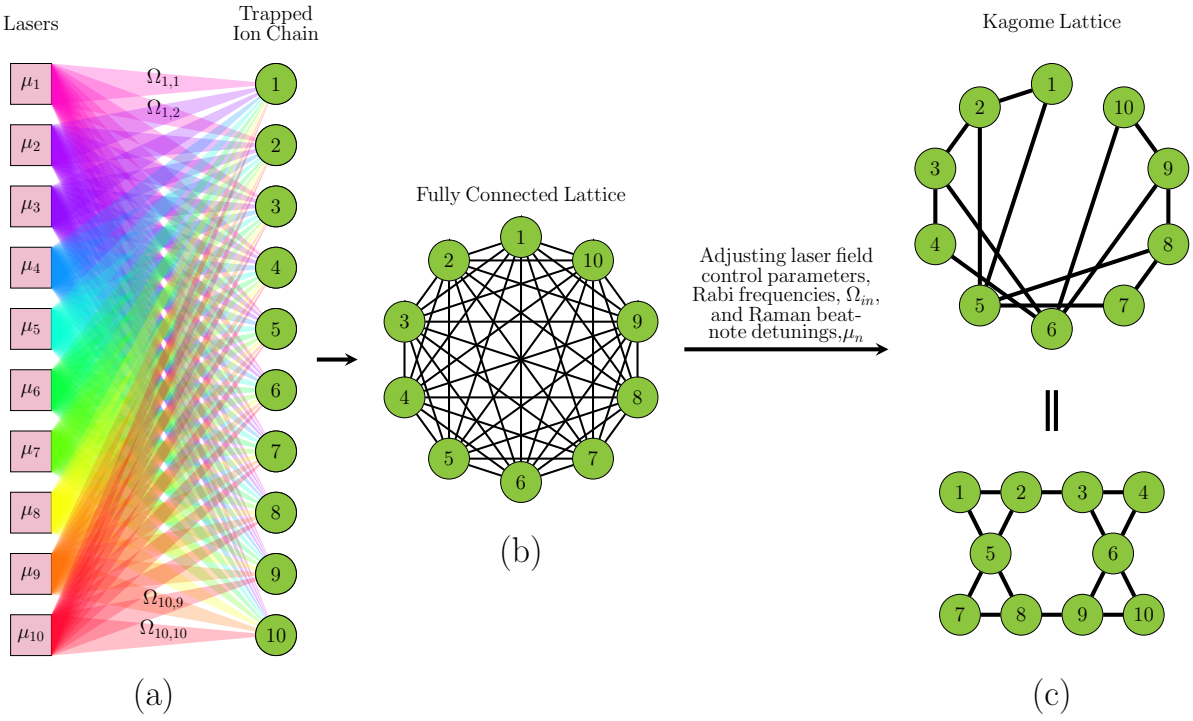


Figure 2.2: **Scheme for arbitrary spin-model simulation on a trapped-ion system.** (a) Diagram of 10 linearly trapped ions individually addressed by 10 bi-chromatic beams, with different frequencies, coming from the left. A counter-propagating (from the right) global beam, omitted from (a), is also applied on the trapped ions. Raman beat-notes, with detuning, μ_n, are generated due to the frequency difference between the individual addressing and global beams. The interaction between the ions is controlled by adjusting the Raman beat-note detunings and Rabi frequencies, Ω_{in}. (b) Generally, we obtain a fully connected lattice with pairwise interaction between all ions. By tuning the control parameters, an arbitrary lattice, e.g. (c) kagome lattice, can be generated.

Notice in Eq. 2.4, by scaling the Rabi frequencies, Ω_{in}, globally by an arbitrary factor, the interaction graph, J_{ij}, scales globally by the square of that factor. Therefore, if we are able to determine the optical field parameters to generate the shape of the interaction graph we desire, characterized by the normalized interaction graph, $\hat{J}_{ij} = J_{ij}/\|J\|$, where $\|J\| = \sqrt{\sum_{i<j} J_{ij}^2}$ is the vector-norm of the interaction graph, we are able to scale the Rabi frequency to match the strength of the target interaction graph.

A caveat with the use of Eq. 2.4 is that direct phonon excitations must be avoided, this implies that the Raman beat-note detunings, μ_n, has to be sufficiently detuned from the phonon mode eigenfrequencies, $\mathbf{w}$, for a particular set of Rabi frequencies, i.e.,

$$|\mu_n - \mathrm{w}_m| \gg \eta_{im}\Omega_{in}. \tag{2.5}$$

To satisfy Eq. 2.5 for the total Rabi frequencies we will be considering, we set the Raman beat-note detunings, μ_n, in the following way,

$$\mu_1 = \omega_1 + 0.1(\overline{\Delta \mathrm{w}}), \tag{2.6}$$

$$\mu_n = \omega_n + 0.1(\mathrm{w}_{n-1} - \mathrm{w}_n), \text{ for } n > 2, \tag{2.7}$$

where $\overline{\Delta \mathrm{w}}$ is the mean separation between neighbouring phonon mode frequencies. Here, the factor of 0.1 is a conservative estimate, and it could be relaxed to allow for the simulation of larger interaction strengths, $\|J\|$, we will elaborate on this further.

2.2 Neural network for programming spin models

To obtain an arbitrary spin model, we look to determine the Rabi frequencies, Ω_{in}, given the Raman beat-note detunings, μ_n, defined by Eq. 2.6 and 2.7, for the normalized interaction graph for the spin model. Given the non-linearity in Eq. 2.4, analytically inverting the relation between the interaction graph and the Rabi frequencies is difficult and maybe impossible. As mentioned before, constrained non-linear optimization has been attempted to determine a solution for a particular interaction graph [18]. In contrast, we developed a more efficient and scalable approach to the problem, approximating the inverse of Eq. 2.4 using the ability of feedforward neural networks (FFNNs) as a universal function approximator. A data-driven approach is employed, a large amount of forward problems are first solved to produce a data set. The data is then fed to the FFNN to train it to approximate the inverse function.

In terms of software, we implemented the neural network using Pytorch, an open source machine learning library [24] for Python. In terms of hardware, we ran the code on a single Nvidia GTX 1060 graphics processing unit (GPU) with 6GB of video random access memory (VRAM).

2.2.1 Neural Network Structure

We adopted the algorithm from previous applications to inverse problems in physics [21] with slight modifications to tailor it to our problem. The structure of the neural network,

shown in Fig. 2.3, is as follows,

- The first layer is the input layer which takes the target normalized interaction graph, $\hat{J}_{ij}^{\text{tar}}$. The number of units in this first layer is $N(N-1)/2$ given by the number of independent elements in $\hat{J}_{ij}^{\text{tar}}$.
- The input layer is fully connected to the hidden layer with 16384 ReLU units. The form of the ReLU unit is shown in Table 1.2. We empirically determined the hidden layer size of 16384, we found that this allowed the neural network to generalize well to unseen data for up to 50 ions. We further investigate the effects of the hidden layer size in Section 2.2.5.
- The hidden layer is fully connected to the output layer which has N^2 linear units. We use these outputs as the Rabi frequencies predicted by the neural network, $\Omega_{in}^{\text{pred}}$. The choice of linear activation is such that the neural network has greater freedom in its prediction of the Rabi frequencies, i.e., it is able to output negative values. These negative Rabi frequencies correspond to a phase difference, between the individual addressing beams and the global beam, of π.

The parameters of the neural network include the weights that connect the input and hidden layers, $W^{(1)}$, and the hidden and output layers, $W^{(2)}$, and the biases of the hidden and output layer, $b^{(1)}$ and $b^{(2)}$ respectively. These are the parameters that are adjusted during the training phase.

During training, we apply a dropout layer between the hidden and output layer that randomly drops 5% of the connections, this helps prevent overfitting in the neural network. Also during training, the outputs of the neural network, the predicted Rabi frequencies, are fed into Eq. 2.4 to generate the normalized interaction graph corresponding to the predicted Rabi frequencies, $\hat{J}_{ij}^{\text{pred}}$. The difference between $\hat{J}_{ij}^{\text{pred}}$ and the target normalized interaction graph, $\hat{J}_{ij}^{\text{tar}}$, is minimized to train the parameters of the neural network.

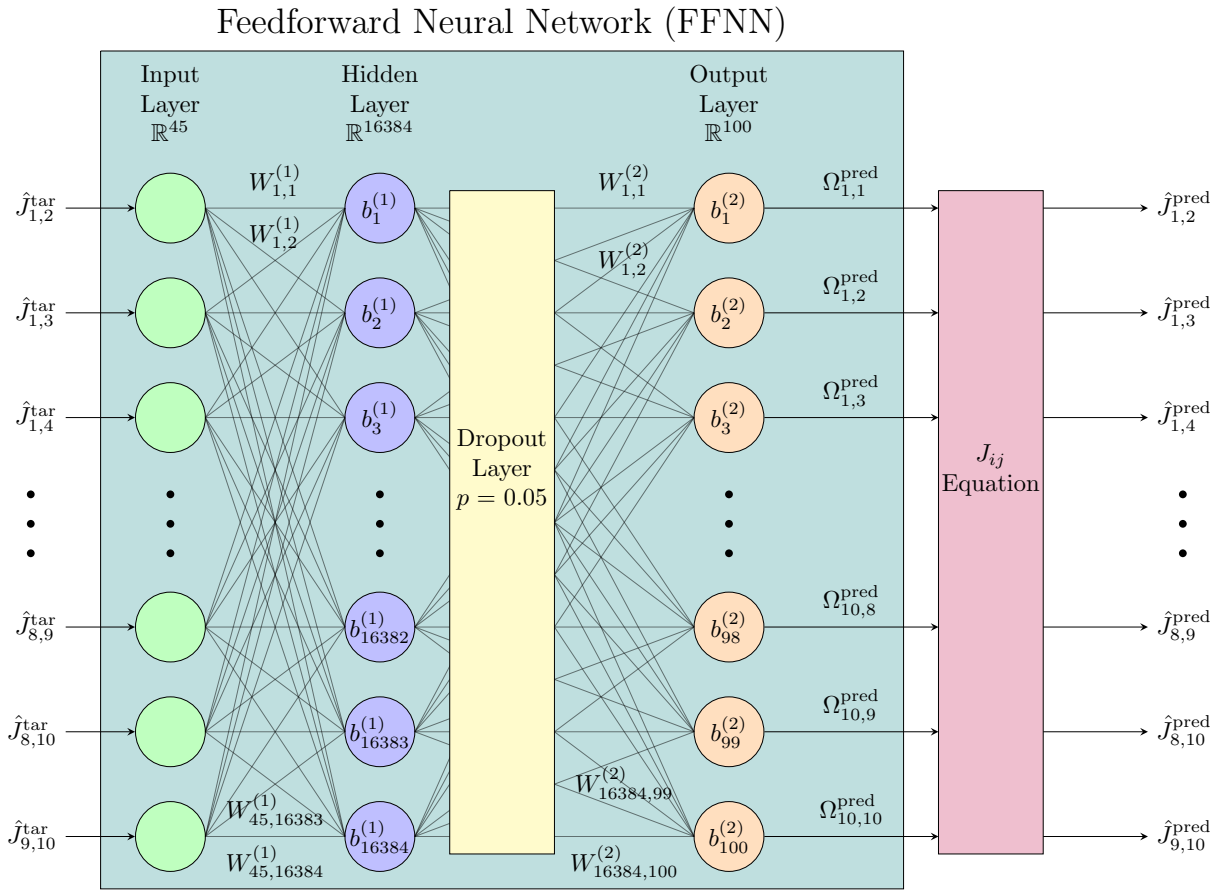


Figure 2.3: **Feedforward neural network for programming spin models.** The feedforward neural network takes the target normalized interaction graph, $\hat{J}^{\text{tar}}_{ij}$, as its inputs and outputs the predicted Rabi frequencies, Ω_{in}. Substituting the predicted Rabi frequencies into Eq. 2.4 and normalizing, we obtain the corresponding neural network programmed normalized interaction graph, $\hat{J}^{\text{pred}}_{ij}$. The full details of the neural network are specified in the text.

2.2.2 Training parameters

The data set used to train and validate the neural network is formed by randomly generating target normalized interaction graphs, $\hat{J}^{\text{tar}}_{ij}$. The procedure is as follows:

1. Randomly generate Rabi frequencies, Ω_{in}, using a continuous uniform distribution

from -1 to 1.

2. Feed the Rabi frequencies into Eq. 2.4 and obtain a target interaction graph, J_{ij}^{tar}.

3. Normalize the target interaction graph.

4. Repeat steps 1-3 to generate the amount of data required for training and validification of the neural network.

The ADAM algorithm [14], a method for stochastic optimization, was utilized to train the neural network parameters (the weights and biases). The optimization was performed on the mean squared error (MSE) loss function between the target normalized interaction graph and neural network programmed interaction graph, $\hat{J}_{ij}^{\text{tar}}$ and $\hat{J}_{ij}^{\text{pred}}$ respectively.

$$C(\hat{J}_{ij}^{\text{pred}}, \hat{J}_{ij}^{\text{tar}}) = \frac{2}{N(N-1)} \sum_{i<j} \left(\hat{J}_{ij}^{\text{pred}} - \hat{J}_{ij}^{\text{tar}} \right)^2 . \tag{2.8}$$

Other relevant hyperparameters for the training of the neural network include: sizes for the training and validation data sets, which are 45000 and 5000 respectively, an initial learning rate of 10^{-3} which is decayed every 5 epochs by 90%. The neural network for trained for a total of 100 epochs.

2.2.3 Numerical results

To evaluate the performance of the neural network, we consider using it to program a trapped-ion quantum spin simulator composed of N ytterbium ions, $^{171}\text{Yb}^{+}$, with Raman beams, propagating along the y-directions, of mean wavelength 355 nm. As the Raman beams are propagating in the y-direction, they only couple to the y-direction phonon modes as is evidenced by Eq. 2.3 and 2.4 when the wave vector difference only has a y-component.

We set the trap frequencies such that the equilibrium configuration of the trapped ions form a linear chain in the z-direction and the y-direction phonon mode eigenfrequencies lie in the range of $2\pi\times1$ MHz to $2\pi\times5$ MHz. Following this, the Raman beat-note detuning are set using Eqs. 2.6 and 2.7. An example is shown for $N = 10$ ions in Fig. 2.4.

For $N = 10$ ions

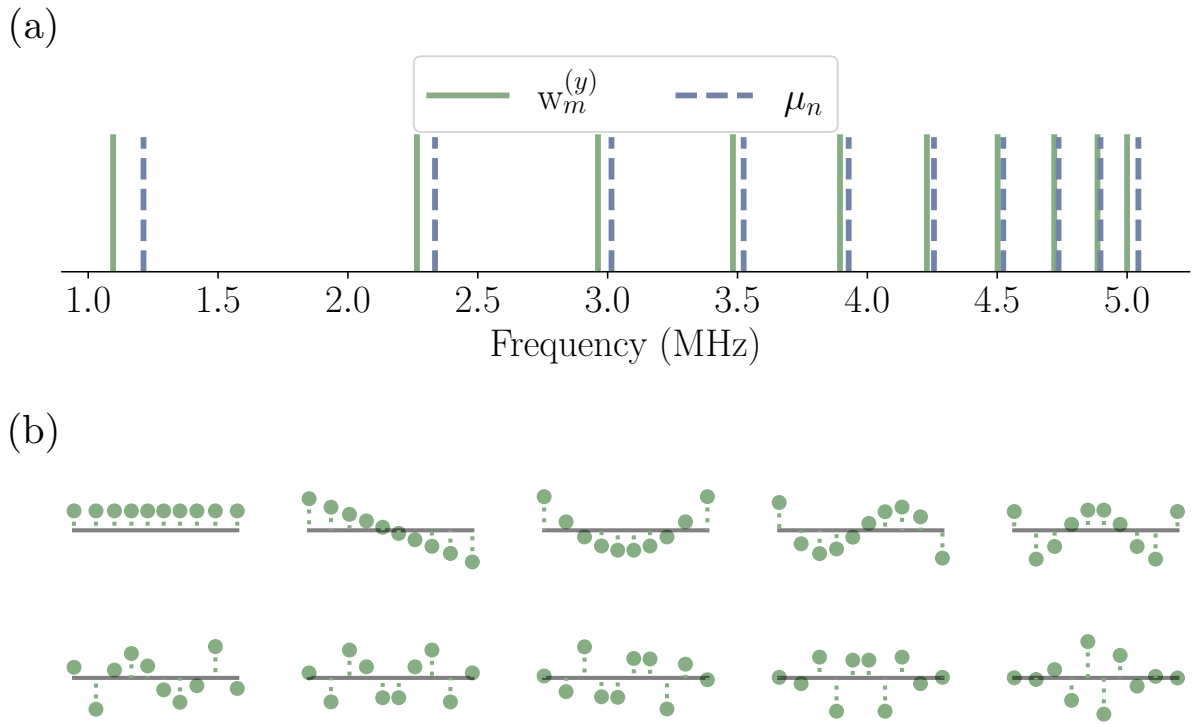


Figure 2.4: **Transverse phonon modes of a trapped-ion system.** The y-direction (a) eigenfrequencies, $\mathbf{w}$, and (b) eigenvectors, $\mathbf{B}$, of a trapped-ion system of 10 ytterbium ions, $^{171}Yb^{+}$, with trap frequencies $\omega_x/2\pi = 6$MHz, $\omega_y/2\pi = 5$MHz and $\omega_z/2\pi = 1.062$MHz. Additionally, the Raman beat-note detunings, μ_n, are also shown in (a).

We first consider a system of $N = 10$ ions. We generate the data for this system and train the neural network using the methods described in Section 2.2.2. Fig. 2.5 shows the loss function as the parameters of the neural network are optimized during training. After 100 epochs, the loss has plateaued indicating that the neural network has converged on a solution for the pseudo-inverse of the problem.

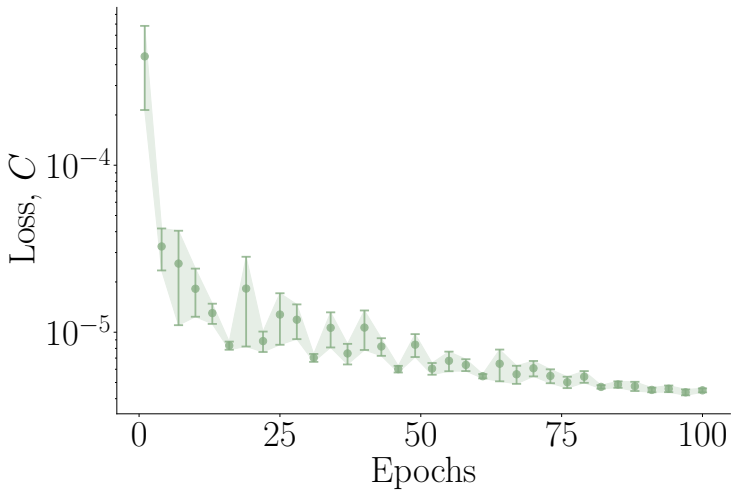


Figure 2.5: **Training results of the neural network for programming spin-models.** The plot shows the loss of the neural network during the training process as more epochs are performed. Notice that we observe good convergence after a 100 epochs.

To further quantify the trained neural network's ability to program the trapped-ion quantum simulator for a target normalized interaction graph, we define a similarity function using the vector inner product between the normalized interaction graphs with form,

$$\mathcal{F}\left(\hat{J}_{ij}^{\text{pred}}, \hat{J}_{ij}^{\text{tar}}\right) = \sum_{i<j} \hat{J}_{ij}^{\text{pred}} \hat{J}_{ij}^{\text{tar}}. \tag{2.9}$$

This similarity function has a range of -1 to 1. When $\mathcal{F} = 1$, the predicted and target normalized interaction graph are identical. When $\mathcal{F} = 0$, the predicted and target normalized interaction graph are "orthogonal", i.e., they are completely different. When $\mathcal{F} = -1$, the predicted and target normalized interaction graph differ by a global negative sign.

We examined the performance of the neural network for programming several regular lattice structures of interest onto the trapped-ion quantum simulator. As an example, Fig. 2.6 shows the results for the Rabi frequencies obtained from the trained neural network when given the two-dimensional Kagome lattice for $N = 10$ ions.

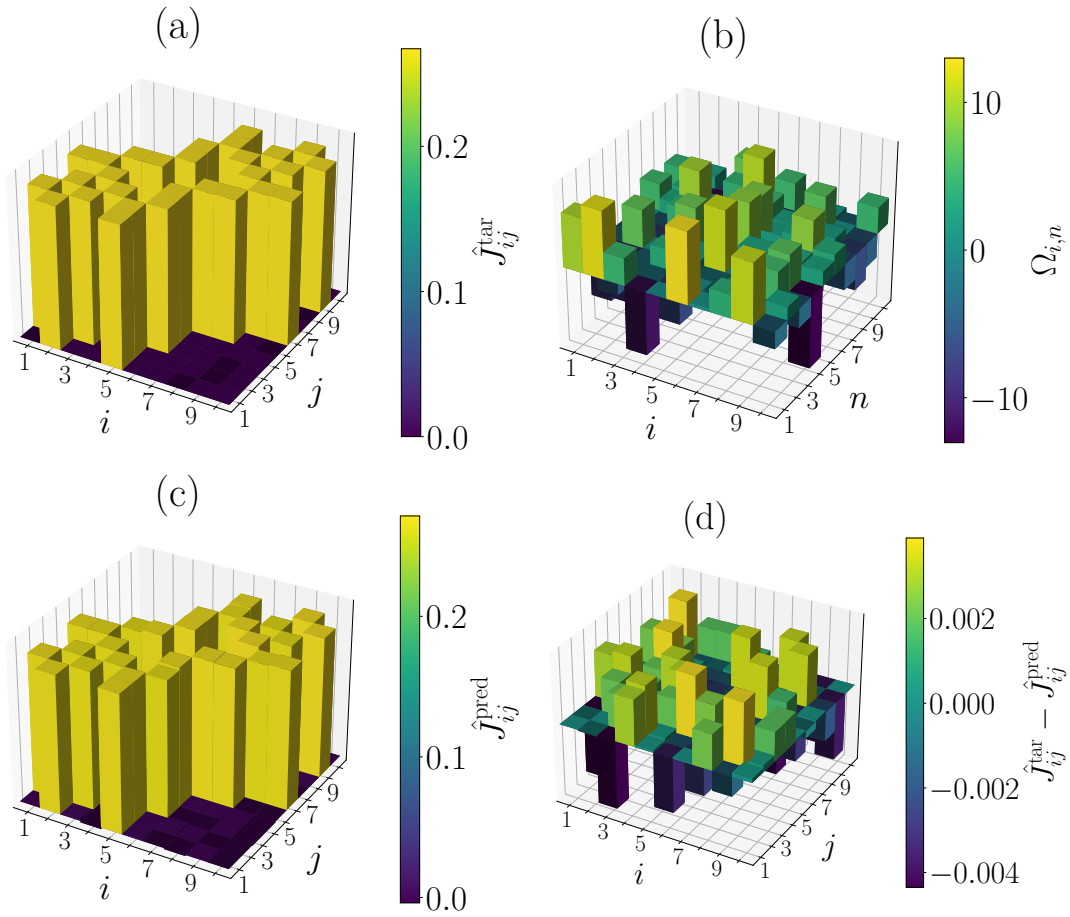


Figure 2.6: **Neural network programmed Kagome lattice spin model.** (a) The target normalized interaction graph of the Kagome lattice on 10 ions, $\hat{J}_{ij}^{\text{tar}}$. (b) The neural network predicted Rabi frequencies, $\Omega_{in}^{\text{pred}}$. (c) The neural network programmed normalized interaction graph, $\hat{J}_{ij}^{\text{tar}}$. The neural network programmed and target normalized interaction graphs agree to a high precision as can be deduced by the (d) difference between them. The similarity is further quantified by a similarity function $\mathcal{F}$, defined in Eq. 2.9, and shown in Table 2.1.

Table 2.1 shows the similarity function for these lattices of interest. The neural network programmed interaction graph was able to achieve similarities, with the target lattice interaction graph, in excess of 0.999.

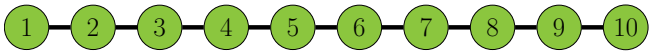
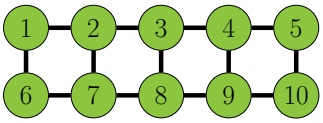
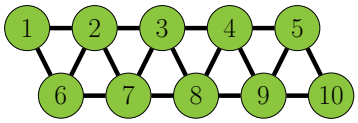
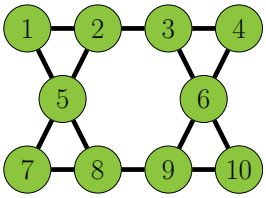
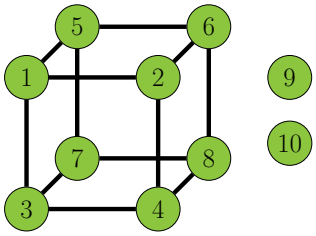
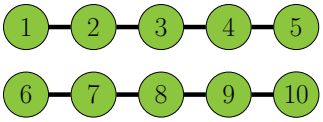
Lattice	Diagram	Similarity, $\mathcal{F}(\hat{J}^{\text{pred}}, \hat{J}^{\text{tar}})$
Chain		0.99988(5)
Square		0.99983(5)
Triangular		0.99989(5)
Kagome		0.99989(5)
Cubic		0.99983(5)
Separated Chains		0.99988(5)

Table 2.1: **Neural network programmed regular lattice spin models.** The table shows the similarity, $\mathcal{F}$, defined in Eq. 2.9, between the neural network programmed and target normalized interaction graph for various lattice geometries. Notice that $\mathcal{F} > 0.999$ for the above lattices, this is sufficiently accurate as it is comparable to the error that arises due to a typical crosstalk of $\epsilon = 0.01$, defined by Eqs. 2.11 to 2.13.

For $N < 50$ ions

In the preceding section, we proved the effectiveness of our technique for $N = 10$ ions. We now explore the scaling of the protocol with larger number of ions. We first adjust the trap

frequencies appropriately, i.e., such that the ions satisfy the aforementioned conditions on its equilibrium configuration and transverse mode frequencies.

In Fig. 2.7 (a), we quantified the performance of the neural network by determining the mean similarity between the neural network programmed normalized interaction graph and the target for the validation data set. For $N < 50$, we observe that $\mathcal{F}$ remains above 0.99 and the decay is a good fit to a function quadratic in the number of ions, $\mathcal{O}(N^2)$.

Next, we characterized the single epoch training time. We observed, in Fig. 2.7 (b), that for N approaching 50 ions, the single epoch training time scales with order, $\mathcal{O}(N^4)$, in the number of ions. The primary limiting factor we observed here is the parallel computation of spin-spin interaction graphs, as it is a memory intensive task. This limitation can be alleviated by an improvement in hardware. A feasible approach to scale the protocol for hundreds of ions would be to employ high performance computing clusters, distributed multi-GPU systems or specialized machine learning hardware, such as tensor processing units.

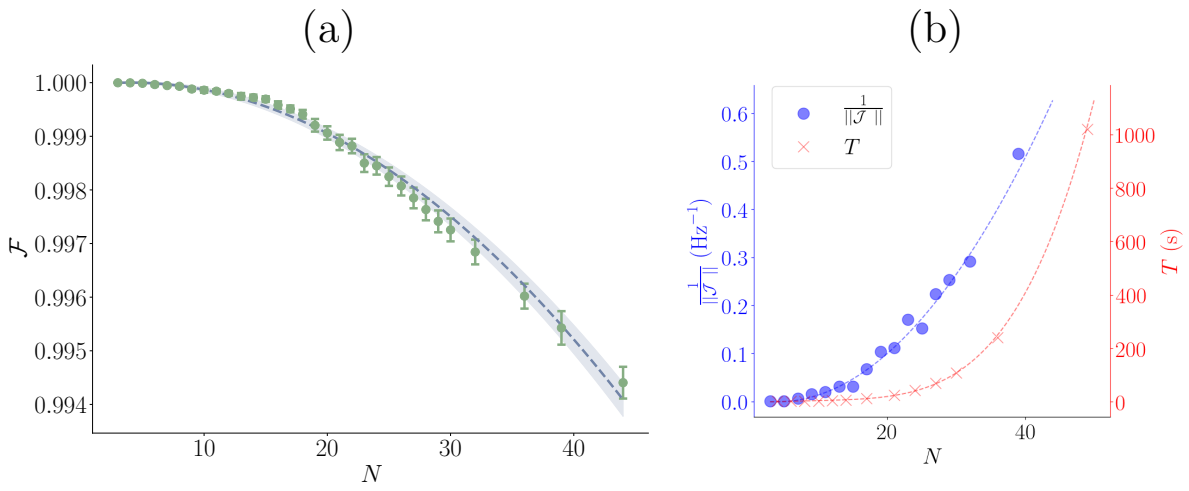


Figure 2.7: **Scaling of the neural network for programming spin-models.** (a) The performance of the neural network measured by the mean similarity between the neural network programmed and target normalized interaction graphs for 200 randomly generated targets. In (a), The dashed line indicates the quadratic fit and the shaded region shows the error bar for the quadratic fit. (b) The single epoch training time (red) of the neural network grows with $\mathcal{O}(N^4)$ and the interaction strength (blue), $||\mathcal{J}||$, of a linear chain for a predetermined total Rabi frequency of, $\sum_{i,n}|\Omega_{i,n}|/2\pi = 1\text{MHz}$, scales inversely with $\mathcal{O}(N^2)$.

2.2.4 Experimental considerations

Above, we established the efficacy of our protocol for programming spin models of arbitrary geometries. However, the magnitude of the interaction has not been taken into consideration. Theoretically, we are able to match an arbitrary interaction strength by globally scaling the Rabi frequencies produced by the neural network. However, experimentally, the finite power of lasers enforces a constraint on the total Rabi frequency available for the experiment, i.e.

$$\sum_{i,n}|\Omega_{i,n}| = \Omega_{\max}. \tag{2.10}$$

In the presence of this constraint, there is an upper bound on the maximum interaction strength achievable. This upper bound is also affected by the amount of ions in the system,

since the power of the laser must be distributed across all the ions. To investigate the scaling of this upper bound with the number of ions, in Fig. 2.7 (b), we program a linear chain spin model onto trapped-ion systems with varying number of ions. We observe that the interaction strength scales inversely with order, $\mathcal{O}(N^2)$, in the number of ions.

Another constraint of concern is given by Eq. 2.5. Satisfaction of the constraint ensures the spin-phonon interactions can be ignored to obtain a pure spin-spin Hamiltonian. In the case of violation, the Hamiltonian of the trapped ion system would not agree with Eq. 2.1 as a spin-phonon term has to be accounted for. The constraint in question is violated for a Rabi frequency in excess of the separation between the Raman beat-note detunings and phonon mode eigenfrequencies. As a result, the Rabi frequencies and therefore interaction strengths attainable on the system are further limited.

The combination of the aforementioned constraints both affect the upper bound on achievable interaction strengths on the system. As such, larger interaction strengths can only be achieved through the relaxation of one of the constraints. Relaxing the first constraint corresponds to obtaining a laser with higher power, this depends on the commercial availability of lasers and may not be a feasible option. Relaxing the second constraint entails selecting Raman beat-note detunings that are closer to the phonon mode phonon mode eigenfrequencies. However, this comes at the expense of additional phonon excitations which lead to errors in the quantum simulation experiment.

In the implementation of an individual addressing scheme on the ions, due to imperfections in the focus of the beam, light leaks to neighbouring ions. This is called cross-talk and is a significant source of error observed when performing individual addressing. We investigate the effect of cross-talk, for the system of $N = 10$ ions, by defining a cross-talk magnitude, ϵ. We model the effect of cross-talk on the Rabi frequencies as follows,

$$\Omega_{i,n}^{\text{cross-talk}} = \Omega_{i,n} + \epsilon\,(\Omega_{i-1,n} + \Omega_{i+1,n}),\ \text{for } 1 < i < 10, \tag{2.11}$$

$$\Omega_{1,n}^{\text{cross-talk}} = \Omega_{1,n} + \epsilon\,\Omega_{2,n}, \tag{2.12}$$

$$\Omega_{10,n}^{\text{cross-talk}} = \Omega_{10,n} + \epsilon\,\Omega_{9,n}. \tag{2.13}$$

Considering a typical crosstalk of 1%, corresponding to $\epsilon = 0.01$, we note that error due to crosstalk is comparable with the error in the machine learning programmed interaction graph. We surmise that our neural network produces sufficiently accurate results for the conditions expected in a realistic experimental situation.

2.2.5 Neural network hyperparameters

In this section, we study the relation between the hyperparameters and performance of the proposed neural network. These hyperparameters include the size of the hidden layer, the dropout ratio for the dropout layer used during the training phase and the amount of training data provided.

From Fig. 2.8 (a) and (b), we observe that an increase in the number of hidden units or the amount of data used to train the neural network results in marginal improvement to the performance of the neural network for $N = 10$ ions. In addition, we see that the neural network's accuracy on the training and validation data sets is similar. This informs us that the neural network does not exhibit overfitting, a major concern in machine learning as it limits the ability of the neural network to generalize to unseen data.

In Fig. 2.8 (c), we see that the dropout layer performs as designed. Raising the ratio of connections dropped during the training phase, closes the gap between the performance of the neural network on the validation and training data sets. However, we discover that this comes at the expense of the neural network's accuracy. As overfitting does not seem to be a concern, the use of a modest dropout ratio of 5% is justified.

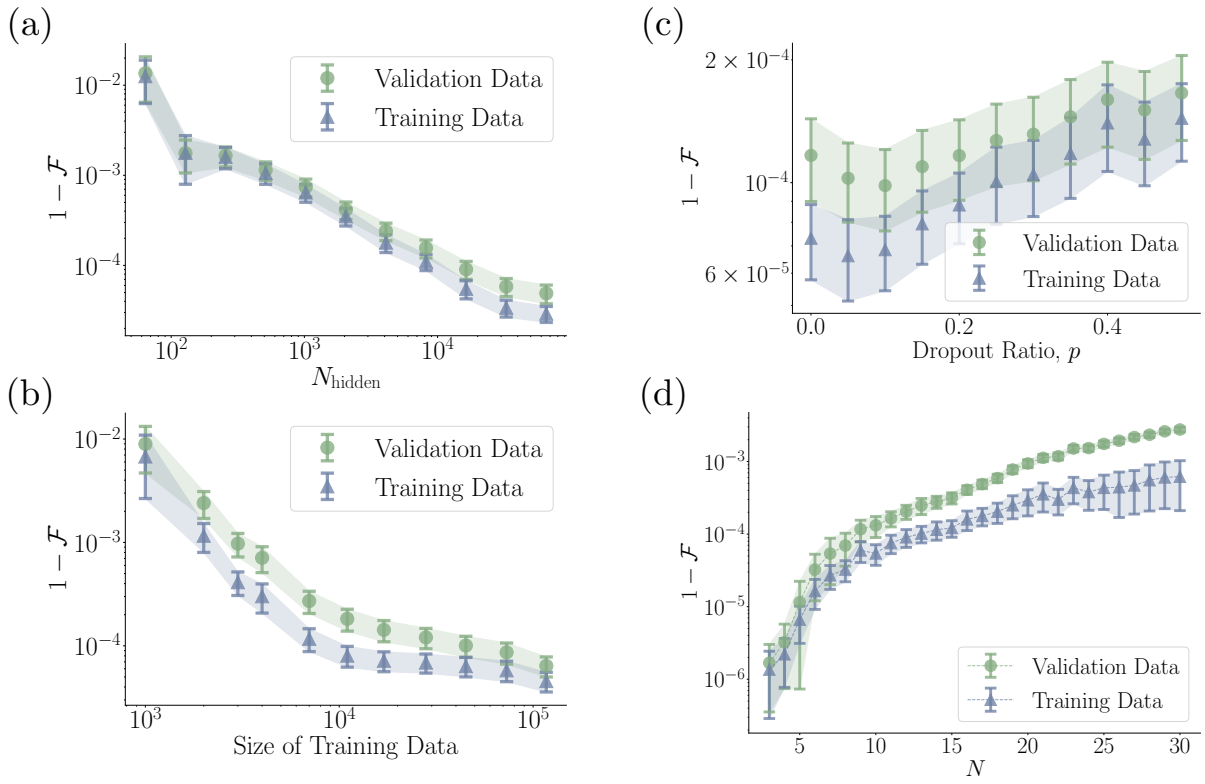


Figure 2.8: **Hyperparameter adjustment of the neural network.** Performance of the neural network, measured by the mean similarity between the neural network programmed and target normalized interaction graphs, for the training (blue) and validation (green) datasets as the hyperparameters, (a) number of hidden units, N_{hidden}, (b) the size of the training dataset and (c) the dropout ratio, p, are adjusted. (d) Comparison between the performance of the neural network on the training and validation datasets as the number of ions, N is varied. It can be deduced that the neural network does not exhibit overfitting for the parameters of the neural network given in Section 2.2.1 and 2.2.2.

In Fig. 2.8 (d), we further look for symptoms of overfitting in systems of up to $N = 30$ ions. We observed that for the smallest system size considered, $N = 3$, there is no sign of overfitting with the settings of the neural network given in Section 2.2.1 and 2.2.2. The accuracy of our neural network, for this smallest system, on the validation and training data sets is within margin of error. However, for larger number of ions, we noticed a discrepancy between the performance of the neural network on the validation and training

data sets. Further improvements to the neural network, such as increasing the size of the training data set and hidden layers, could possibly reduce this discrepancy.

2.3 Conclusion

In this chapter, we discussed the mechanism used by a trapped-ion system to perform the simulation of a quantum spin model via the application of an optical field. We proposed the use of a feedforward neural network to program an arbitrary spin model onto such a trapped-ion quantum simulator. This protocol, unlike traditional non-linear optimization routines, generates a pseudo inverse function for Eq. 2.4. This enables the quick determination of the control parameters for interaction graphs that were not previously encountered during training. This simple machine learning architecture performs well for systems of up to $N = 50$ ions on a consumer grade GPU workstation. By employing high performance computing clusters, distributed multi-GPU systems, or specialized machine learning hardware such as tensor processing units, the protocol could be further improved and scaled up to hundreds of ions.

Chapter 3

Optical tweezer manipulation of the phonons of a trapped-ion system

As demonstrated in the previous chapter, the phonon modes that arise in a trapped-ion system play a critical role in quantum information processing experiments. Therefore, the ability to manipulate the structure of the phonon modes opens up new possibilities for the trapped-ion platform [25, 26]. Implementing additional control over the phonon modes requires individual control over the trap strength of each ion. However, this level of control is absent in conventional radio-frequency traps.

On the other hand, optical tweezers, due to the AC Stark effect, are capable of inducing a localized trapping potential. These optical tweezers have been shown to enable precise manipulation in various quantum systems [26–31]. Of particular relevance, optical tweezers have been demonstrated as an alternative method for trapping ions [32–34].

This chapter consist of work accepted for publication in the journal, Physical Review A [2]. In this chapter, we study the shift in the electronic energy levels of an ion due to the introduction of an optical tweezers focused onto the ion. Also, we investigate how the introduction of optical tweezers in the trapped-ion system affects the phonon modes of the system. Furthermore, we introduce the algorithms we developed to program the collective phonon modes of the trapped-ion system via the precise tuning of the optical tweezer parameters. Finally, we study possible extensions of the utility of trapped-ion system afforded to us by the additional control over the phonon modes. Specifically, we look to enhance the inter-species phonon mode coupling in a multi-species trapped-ion system.

3.1 Effect of optical tweezers on an ion

Consider a simplified model of an ion where the ion is approximated as a simple two-level system. Let the ground state and excited state of the ion be denoted by $|g\rangle$ and $|e\rangle$ respectively. Suppose the excited state and the ground state are separated by an energy, $\hbar\omega_a$. Consider introducing an optical field with an electric field, $\mathbf{E} = E_0 \cos(\omega_l t)\boldsymbol{\xi}$, onto the ion. Let the Fock state of the photon field be $|n\rangle$. The Hamiltonian of the system would take the form of a Jaynes-Cumming Hamiltonian:

$$\hat{H} = \frac{\hbar\omega_a}{2}\hat{\sigma}_z + \frac{\hbar\omega_l}{2}\hat{a}^\dagger\hat{a} - \frac{\langle\hat{\mu}\rangle E_0}{2}\left(\hat{a}^\dagger + \hat{a}\right)(\hat{\sigma}_+ + \hat{\sigma}_-) \tag{3.1}$$

Here, $\hat{\sigma}_i$ are the Pauli operators, $\hat{a}^\dagger$ and $\hat{a}$ are the creation and annihilation operators of the photon field and $\langle\hat{\mu}\rangle$ is the expectation value of the electric dipole moment of the electron.

Define the Rabi frequency as,

$$\Omega = \frac{-\langle\hat{\mu}\rangle E_0}{\hbar}, \tag{3.2}$$

and rewrite $\hat{H}$ as:

$$\hat{H} = \frac{\hbar\omega_a}{2}\hat{\sigma}_z + \hbar\omega_l\hat{a}^\dagger\hat{a} + \frac{\hbar\Omega}{2}\left(\hat{a}^\dagger + \hat{a}\right)(\hat{\sigma}_+ + \hat{\sigma}_-). \tag{3.3}$$

Consider moving to the interaction picture, using,

$$\hat{H}_0 = \frac{\hbar\omega_a}{2}\hat{\sigma}_z + \hbar\omega_l\hat{a}^\dagger\hat{a}, \tag{3.4}$$

$$\hat{H}_1 = \frac{\hbar\Omega}{2}\left(\hat{a}^\dagger + \hat{a}\right)(\hat{\sigma}_+ + \hat{\sigma}_-), \tag{3.5}$$

and defining:

$$\hat{H}_{\text{int}} = \hat{H}_0 + \exp\left(\frac{i\hat{H}_0 t}{\hbar}\right)\hat{H}_1\exp\left(-\frac{i\hat{H}_0 t}{\hbar}\right). \tag{3.6}$$

Using the following relations,

$$\exp\left(\frac{i\omega_a t\hat{\sigma}_z}{2}\right)\hat{\sigma}_\pm\exp\left(\frac{-i\omega_a t\hat{\sigma}_z}{2}\right) = \exp(\pm i\omega_a t)\hat{\sigma}_\pm, \tag{3.7}$$

$$\exp\left(i\omega_l t\hat{a}^\dagger\hat{a}\right)\hat{a}^\dagger\exp\left(-i\omega_l t\hat{a}^\dagger\hat{a}\right) = \exp(i\omega_l t)\hat{a}^\dagger, \tag{3.8}$$

$$\exp\left(i\omega_l t\hat{a}^\dagger\hat{a}\right)\hat{a}\exp\left(-i\omega_l t\hat{a}^\dagger\hat{a}\right) = \exp(-i\omega_l t)\hat{a}, \tag{3.9}$$

we obtain the following form for the Hamiltonian in the interaction picture,

$$\hat{H}_{\text{int}} = \frac{\hbar\omega_a}{2}\hat{\sigma}_z + \hbar\omega_l\hat{a}^\dagger\hat{a} + \frac{\hbar\Omega}{2}\Big(\exp(i(\omega_l+\omega_a)t)\hat{a}^\dagger\hat{\sigma}_+ + \exp(i(\omega_l-\omega_a)t)\hat{a}^\dagger\hat{\sigma}_- + \exp(i(-\omega_l+\omega_a)t)\hat{a}\hat{\sigma}_+ + \exp(i(-\omega_l-\omega_a)t)\hat{a}\hat{\sigma}_-\Big). \tag{3.10}$$

Assuming that $\omega_l + \omega_a \gg \omega_l - \omega_a$, we can use the rotating wave approximation (RWA) to disregard the fast oscillating terms with frequency $\omega_l + \omega_a$, this simplifies the Hamiltonian to,

$$\hat{H}_{\text{int}} = \frac{\hbar\omega_a}{2}\hat{\sigma}_z + \hbar\omega_l\hat{a}^\dagger\hat{a} + \frac{\hbar\Omega}{2}\left(\exp(i(\omega_l-\omega_a)t)\hat{a}^\dagger\hat{\sigma}_- + \exp(i(-\omega_l+\omega_a)t)\hat{a}\hat{\sigma}_+\right). \tag{3.11}$$

Transforming back to the Schrödinger picture we get,

$$\hat{H} = \frac{\hbar\omega_a}{2}\hat{\sigma}_z + \hbar\omega_l\hat{a}^\dagger\hat{a} + \frac{\hbar\Omega}{2}\left(\hat{a}^\dagger\hat{\sigma}_- + \hat{a}\hat{\sigma}_+\right). \tag{3.12}$$

Consider the submanifold of states $|g,n\rangle$ and $|e,n-1\rangle$ as these two states are coupled by the introduction of the electric field, this is shown in Eq. 3.12 as terms $\hat{a}^\dagger\hat{\sigma}_-$ and $\hat{a}\hat{\sigma}_+$. In this submanifold, the hamiltonian has the following form,

$$\hat{H} = \frac{\hbar}{2}\begin{bmatrix} -\omega_a + (2n+1)\omega_l & \Omega \\ \Omega & \omega_a + (2n-1)\omega_l \end{bmatrix}. \tag{3.13}$$

Suppose we define $\delta = \omega_l - \omega_a$,

$$\hat{H} = \frac{\hbar}{2}\begin{bmatrix} \delta & \Omega \\ \Omega & -\delta \end{bmatrix} + n\hbar\omega_l\mathbb{I}, \tag{3.14}$$

where, $\mathbb{I}$ is the identity matrix.

The eigenvalues of $\hat{H}$ in this submanifold are,

$$\lambda_\pm = \pm\frac{\hbar\delta}{2}\sqrt{1+\frac{\Omega^2}{\delta^2}} + n\hbar\omega_l. \tag{3.15}$$

In the regime where $\Omega \ll \delta$,

$$\lambda_\pm = \pm\frac{\hbar\delta}{2}\left(1+\frac{\Omega^2}{2\delta^2}\right) + n\hbar\omega_l. \tag{3.16}$$

Therefore, the change in energy, also known as the AC Stark shift, $\Delta\mathcal{E}$, of the ground and excited state in this submanifold due to interaction with the electric field is,

$$\Delta\mathcal{E}_g = \Delta\mathcal{E} = \frac{\hbar\Omega^2}{4\delta}, \tag{3.17}$$

$$\Delta\mathcal{E}_e = -\Delta\mathcal{E} = -\frac{\hbar\Omega^2}{4\delta}. \tag{3.18}$$

Notice from Eq. 3.2, the Rabi frequency, Ω, is proportional to the electric field amplitude, E_0. Therefore, Ω^2 and thus the AC Stark shift, $\Delta\mathcal{E}$, would both be proportional to the intensity of the optical field, I. Let $\bar{\Omega}^2$ be the proportionality factor between Ω^2 and I.

Consider a Gaussian optical tweezer, i.e., focused laser beam, propagating along the x-direction, its intensity has the following spatial dependence,

$$I(\mathbf{r}) = I_0 \left(\frac{\sigma_0}{\sigma(x)}\right)^2 \exp\left(-\frac{2(y^2+z^2)}{\sigma^2(x)}\right). \tag{3.19}$$

Here, $\sigma(x) = \sigma_0\sqrt{1+(x/x_R)^2}$ is the spot size, σ_0 is the beam waist, $x_R = \pi\sigma_0^2 n/\lambda$ is the Rayleigh range, λ is the wavelength of the optical tweezer, n is the refractive index of the optical medium, and $I_0 = 2P/\pi\sigma_0^2$ is the peak intensity of the optical tweezer of power P.

By applying the optical tweezer onto the ion, the ion experiences a spatially dependent Stark shift which is an additional potential on the ion of the form,

$$\phi^{\text{opt}}(\mathbf{r}) = \Delta\mathcal{E}(\mathbf{r}) = \frac{\hbar\bar{\Omega}^2}{4\delta} I(\mathbf{r}). \tag{3.20}$$

Here, ϕ^{opt} is the optical potential, i.e., the potential generated by applying an optical tweezer on an ion.

By applying the harmonic approximation to the optical potential on the ion at the focus of the Gaussian optical tweezer, i.e., $\mathbf{r} = 0$, we approximate the optical potential as either a harmonic trapping or anti-trapping potential of the following form,

$$\phi^{\text{opt}}(\mathbf{r}) \approx s^{\text{opt}} \frac{M}{2} \left[(\omega_x^{\text{opt}})^2 x^2 + (\omega_y^{\text{opt}})^2 y^2 + (\omega_z^{\text{opt}})^2 z^2\right]. \tag{3.21}$$

where,

$$\omega_x^{\text{opt}} = \sqrt{\left|\frac{\chi I_0 \lambda^2}{2\pi^2 \sigma_0^4 n^2 M}\right|}, \tag{3.22}$$

$$\omega_y^{\text{opt}} = \omega_z^{\text{opt}} = \sqrt{\left|\frac{\chi I_0}{\sigma_0^2 M}\right|}, \tag{3.23}$$

$$\chi = \frac{\hbar \bar{\Omega}^2}{\delta}, \tag{3.24}$$

$$s^{\text{opt}} = -\text{sign}(\delta). \tag{3.25}$$

Here, $s^{\text{opt}} = \pm$ determines if the potential is trapping or anti-trapping and $\omega_\alpha^{\text{opt}}$ is the effective trap strength of the optical tweezer in the α-direction.

In the following sections, we apply the above calculation for a physical ion of $^{171}\text{Yb}^+$, with electronic structure shown in Fig. 1.1, we consider the $S_{1/2}$ manifold as our ground states, $|g\rangle = \left|S_{1/2}, F = 0, 1, m_F = 0, \pm 1\right\rangle$, and the $P_{1/2}$ manifold as our excited state, $|e\rangle = \left|P_{1/2}, F = 0, m_F = 0\right\rangle$.

3.2 Hybrid trapped-ion system

In this section, we introduce additional optical tweezers, propagating along x, focused precisely onto each ion in the conventional trapped-ion apparatus shown in Fig. 1.2. A schematic of this hybrid system is shown in Fig. 3.1. Since the intensity of light away from the focus of the optical tweezer is suppressed exponentially in the z-direction, by assuming the beam waist is sufficiently smaller than the inter-ion spacing, we can neglect the effects of the optical tweezers on neighbouring ions. The potential of such a system has the following form:

$$\phi^{\text{hyb}} = \phi^{\text{conv}} + \sum_i \phi_i^{\text{opt}}(\mathbf{r}_i - \mathbf{r}_i^*), \tag{3.26}$$

$$= \phi^{\text{Coulomb}} + \phi^{\text{DC+RF}} + \sum_i \phi_i^{\text{opt}}(\mathbf{r}_i - \mathbf{r}_i^*) \tag{3.27}$$

where, ϕ^{Coulomb} (Eq. 1.2) and $\phi^{\text{DC+RF}}$ (Eq.1.3) are the Coulomb potential and the harmonic pseudopotential generated by the DC and RF electrodes respectively. ϕ_i^{opt} is the optical

potential generated by the i-th optical tweezer focused onto the i-th ion at its equilibrium position, $\mathbf{r}_i$, and it has the form detailed in Eqs. 3.19 and 3.20 with controllable power.

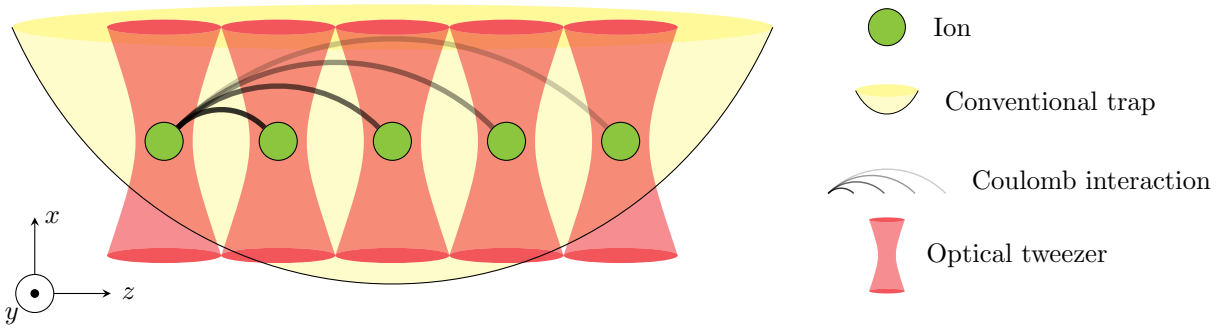


Figure 3.1: **Schematic of the potential of the proposed hybrid trapped-ion system.** Additional optical tweezers, with controllable power, are introduced into a conventional radio-frequency trap (Fig. 1.2). Each optical tweezer is focused onto one of the $N = 5$ ions trapped.

Under the assumption that the optical tweezers induce a trapping (or weakly anti-trapping) potential and are precisely focused onto the ions, we can neglect the changes to the equilibrium position of the ions.

Under the harmonic approximation for the optical potential (Eq. 3.21), we can define the total trapping strength of the hybrid trap in the α-direction on ion i as:

$$\omega_{\alpha,i}^{\text{hyb}} = \sqrt{\omega_{\alpha,i}^2 + s_i^{\text{opt}} \left(\omega_{\alpha,i}^{\text{opt}}\right)^2}. \tag{3.28}$$

where, $\omega_{\alpha,i}$ is given by Eqs. 1.4 to 1.6, $\omega_{\alpha,i}^{\text{opt}}$ and s_i^{opt} are given by Eqs. 3.22 to 3.25.

In order to determine the phonon modes of the hybrid system, we diagonalize the $\mathbf{A}$-matrix defined in Eq. 1.7 substituting ϕ^{conv} for ϕ^{hyb}.

In taking the Hessian of a harmonic potential, it is trivial to deduce that the off-diagonal elements would be zero, whereas, the diagonal elements would be the square of the trapping strengths. As such, it should be noted that the off-diagonal elements of the $\mathbf{A}$-matrix are completely determined by the Coulomb potential, whereas, the diagonal elements depend on each term in ϕ^{hyb}. From Eqs. 1.4 to 1.6, the RF and DC electrodes induce identical

trapping potentials for ions of the same charge and mass. In contrast, by controlling the power of the optical tweezer on each ion, we acquire the ability to independent vary the trapping potential at each site. This implies the diagonal elements of the **A**-matrix can be modified independently. Consequently, we gain limited, not arbitrary (shown in Appendix B), control over the phonon modes of the system via the optical tweezers.

3.3 Effect of optical tweezers on phonon modes.

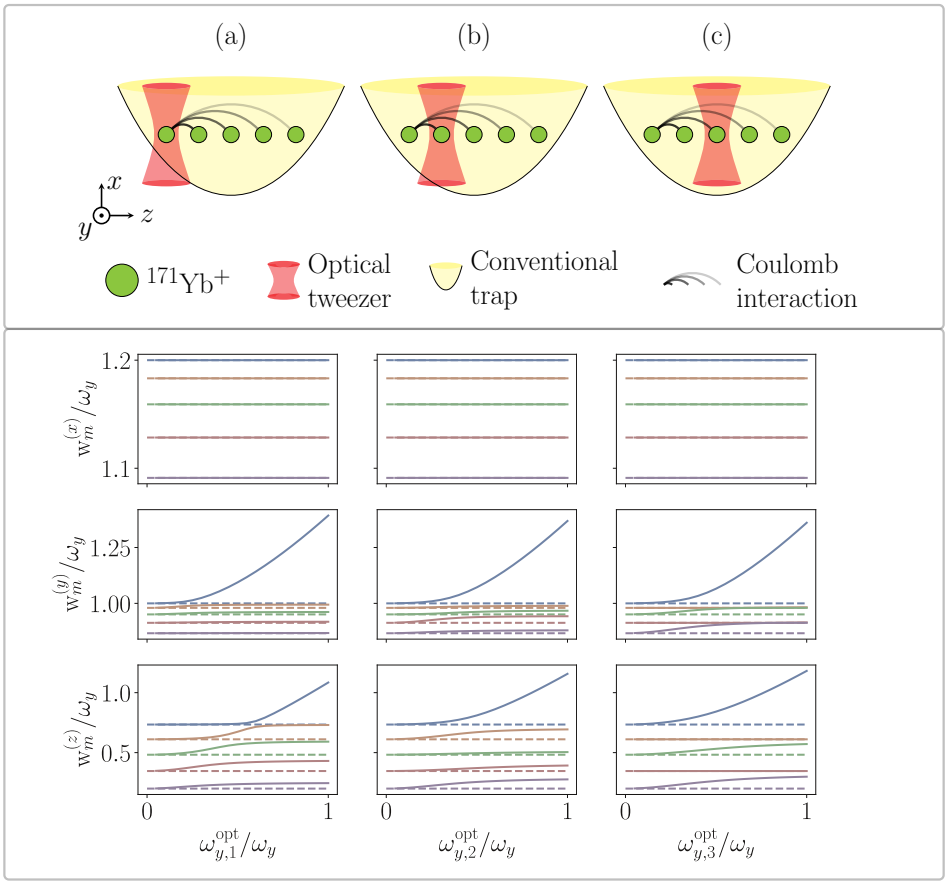


Figure 3.2: **Effect of an optical tweezer on the phonon mode eigenfrequecies of a trapped-ion system.** (Top panel) An additional optical tweezer is applied onto ion-i of the conventional trapped-ion system. The three independent locations for the optical tweezer are shown in (a) through (c). Parameters of the system are described in the text. (Bottom panel) The phonon mode eigenfrequencies of the trapped-ion system in x, y and z as the optical trap strength is varied, the colors in the plot represent different mode number and the dotted lines represent the conventional phonon mode eigenfrequencies. Notice, the x-mode frequencies are barely affected by the optical tweezers and the y and z-mode frequencies increase with optical trap strength.

In this section, we investigate the phonon modes of the hybrid trap proposed in the previous section, where only a single optical tweezer is switched on. We consider a system of $N = 5$ $^{171}Yb^{+}$ ions, where the ions have conventional trap frequencies of, $\omega_x/2\pi = 1.2$ MHz, $\omega_y/2\pi = 1$ MHz and $\omega_z/2\pi = 0.2$ MHz. We then introduce an optical tweezer on ion-i, with a wavelength of $\lambda = 375$ nm and beam waist of $\sigma_0 = 1$ μm, and vary the power, P. By varying power, we locally modify the trap strength on ion-i in accordance to Eq. 3.28. Due to the reflection symmetry of the trapped-ion chain about the center of the system, there are only three independent locations that would yield different behaviours of the phonon modes.

Fig. 3.2 shows there is no appreciable change to the x-direction phonon mode eigenfrequencies. This is consistent with expectations as for the Gaussian optical tweezer, the intensity decays according to an inverse quadratic relation along the direction of propagation (x), whereas along the directions transverse to the propagation (y, z), the intensity decays exponentially. Consequently, the optical tweezer induces a much weaker confinement (Eq. 3.22) in the x-direction compared to in the y and z-directions, which experiences similar confinement due to the circular symmetry of the Gaussian beam (Eq. 3.23). A weak optical confinement in the x-direction implies the change to the $\mathbf{A}$-matrix is minuscule, therefore, the phonon modes do not experience a significant modification.

In contrast, the y and z-direction phonon mode frequencies generally increase with increasing optical tweezer strength except in the case where ion-i does not participate in the phonon mode. An example is shown in Fig. 3.2(c) and Table 3.1 where the optical tweezer is applied on the center ion. As the center ion does not contribute to the tilt mode and fourth mode (shown in Table 1.1), the optical tweezer on the center ion is incapable of modifying these modes.

Additionally, it can be deduced from Fig. 3.2, that in the limit where $\omega_{y,i}^{\text{opt}} \approx \omega_y$, the i-th ion effectively decouples from the rest of the system. This decoupling effect is clearly demonstrated in Fig. 3.2, where the highest mode is a localized mode for ion-i, i.e., the eigenvector only involves ion-i, whereas, in the remaining modes ion-i is not a participant.

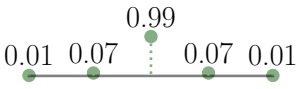
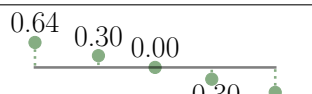
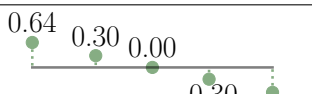
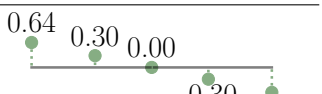
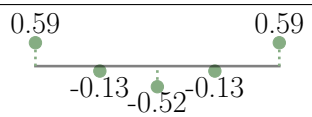
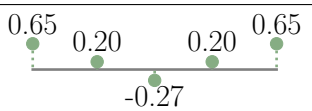
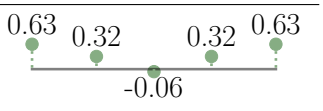
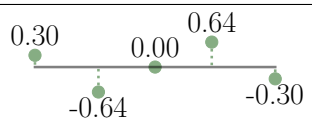
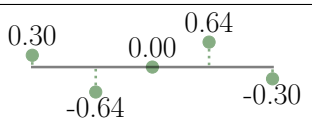
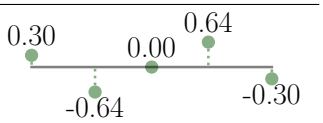
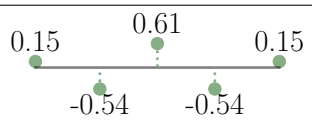
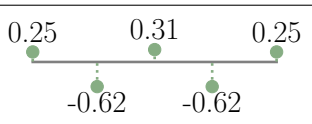
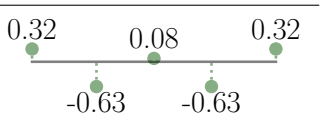
$\omega_{y,3}^{\mathrm{opt}}/\omega_y$	0.25	0.5	1
$\mathrm{B}_{i1}^{(y)}$			
$\mathrm{B}_{i2}^{(y)}$			
$\mathrm{B}_{i3}^{(y)}$			
$\mathrm{B}_{i4}^{(y)}$			
$\mathrm{B}_{i5}^{(y)}$			

Table 3.1: **Effect of an optical tweezer on the phonon mode eigenvectors of a trapped-ion system.** Here, the optical tweezer is applied on the center ion and we show the y-mode eigenvectors for the trapped-ion system. The numbers in the eigenvector plots correspond to normalized vectors that quantify the relative participation of each ion in the phonon modes. Notice, as the optical trap strength is increased we observe the following: In the first mode, the participation of the center ion increases, whereas, the participation of the other ions decrease. For the third and fifth modes, the converse is true. For the second and fourth modes, the optical tweezer has no effect on the modes as the center ion does not participate in these modes.

3.4 Programmable control of Phonon modes

As demonstrated in the previous section, the introduction of optical tweezers modifies the trapping potential and results in changes to the phonon mode structure (eigenfrequencies and eigenvectors) of the trapped-ion system. In this section, we introduce the protocols we developed to enable the programmable control of the phonon mode eigenfrequencies or

eigenvectors.

Without loss of generality, we are able to focus on a particular axis $\alpha \in \{x, y, z\}$, as such, we will drop the α index going forth.

3.4.1 Eigenfrequencies control

Statement of problem

A statement of the phonon mode eigenfrequencies control problem is as follows:

- Determine the required parameters on the optical tweezers $\left(\omega_i^{\text{opt}}, s_i^{\text{opt}}\right)$ such that the trapped-ion system has a target set of phonon mode eigenfrequencies, $\mathbf{w}^{\text{tar}}$.

Or equivalently, the problem could be mathematically formulated as such:

- Determine an $\mathbf{A}$-matrix that satisfies the following constraints simultaneously:

$$\mathrm{A}_{ij} = \sum_m \mathrm{B}_{im} \left(\mathrm{w}_m^{\text{tar}}\right)^2 \mathrm{B}_{jm}, \text{ where } \mathbf{B} \text{ is an arbitrary unitary matrix.} \tag{3.29}$$

$$\mathrm{A}_{ij} = \mathrm{A}_{ij}^{\text{conv}}, \ \forall \ i \neq j. \tag{3.30}$$

  Here, $\mathbf{A}^{\text{conv}}$ is the $\mathbf{A}$-matrix corresponding to the conventional system. Constraint 1 (Eq. 3.29) ensures that the $\mathbf{A}$-matrix has the target eigenfrequencies, $\mathbf{w}^{\text{tar}}$, whereas, constraint 2 (Eq. 3.30) ensures that the $\mathbf{A}$-matrix is realisable with the hybrid trap, i.e., only the diagonal elements of $\mathbf{A}$-matrix is modified compared to the conventional counterpart.

From the mathematical formulation of the problem, we can deduce that: (1) a solution is not guaranteed to exist and (2) solutions may not be unique. We derive these properties in Appendix B.

Iterative diagonalization algorithm (IDA)

In the field of optics, there is the technique of holography. Holography allows for the precision engineering of optical wave fronts, enabling full control over the beam profile. To

program such a hologram, an iterative Fourier transform algorithm (IFTA), also known as the Gerchberg–Saxton algorithm, was developed [35].

In IFTA the goal is to determine the phase profile of a complex wave front with intensity measurements on two planes, the source and target plane. Distilling the key characteristic of the problem, we obtain:

- There are two matrices of interest, in IFTA, these matrices represent the source and target planes.
- The matrices are related by a unitary, in IFTA, this unitary is the Fourier transform.
- There is a constraint on each matrix, in IFTA, the amplitude of the complex wavefront at the source and target planes have been determined by intensity measurements.

Our problem of eigenfrequencies control for the hybrid system exhibits all of the above characteristics. The two matrices that are of interest are the $\mathbf{A}$-matrix and its corresponding eigenvalue matrix. As the $\mathbf{A}$-matrix is symmetric, the diagonalization transform that relates the two matrices is a unitary transform. Additionally, there exist a constraint on each of these matrices, these constraints are embodied by Eq. 3.29 and 3.30. Therefore, in the same spirit as IFTA, we propose an iterative diagonalization algorithm (IDA), a flowchart of which is shown in Fig. 3.3, that would be a key element to solving the problem.

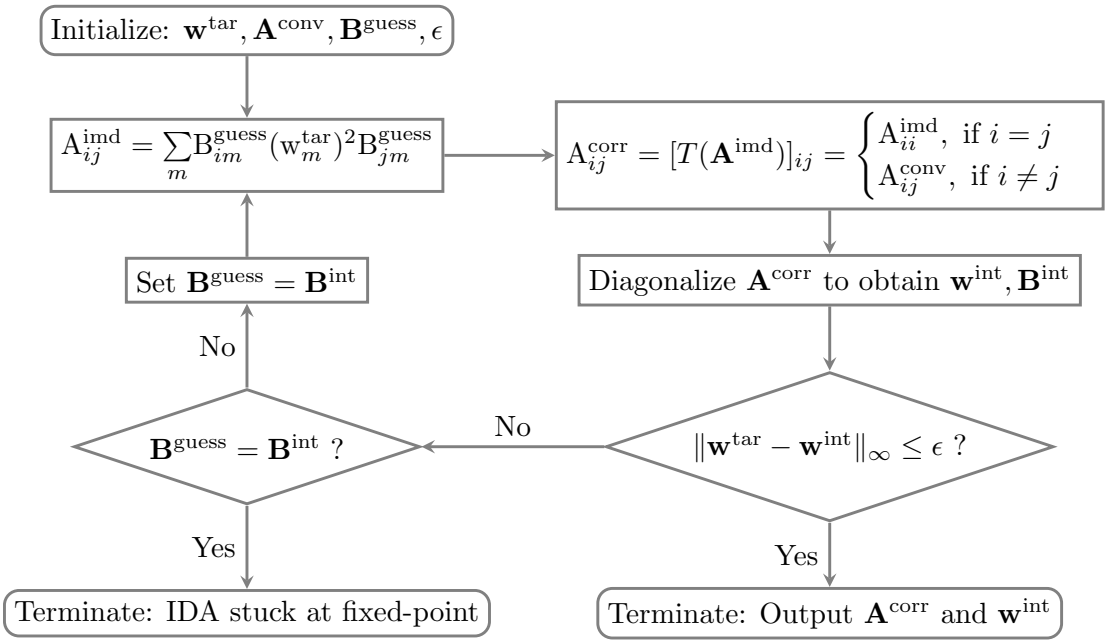


Figure 3.3: **Flowchart for iterative diagonalization algorithm (IDA).**

To initialize IDA, we specify the following parameters: the conventional $\mathbf{A}$-matrix, $\mathbf{A}^{\text{conv}}$, the target eigenfrequencies, $\mathbf{w}^{\text{tar}}$, an initial-guess eigenvector matrix, $\mathbf{B}^{\text{guess}}$, and a desired accuracy, ϵ. IDA proceeds in the following order:

1. Construct an intermediate $\mathbf{A}$-matrix, $\mathbf{A}^{\text{imd}}$:

$$\mathrm{A}_{ij}^{\text{imd}} = \sum_m \mathrm{B}_{im}^{\text{guess}} \left(\mathrm{w}_m^{\text{tar}}\right)^2 \mathrm{B}_{jm}^{\text{guess}}. \tag{3.31}$$

   By construction, $\mathbf{A}^{\text{imd}}$ satisfies constraint 1 (Eq. 3.29). However, it may not satisfy constraint 2 (Eq. 3.30).

2. Correct the off-diagonal elements of $\mathbf{A}^{\text{imd}}$, represented by operator T, to obtain the corrected $\mathbf{A}$-matrix, $\mathbf{A}^{\text{corr}}$.

$$\mathrm{A}_{ij}^{\text{corr}} = \left[T(\mathbf{A}^{\text{imd}})\right]_{ij} = \begin{cases} \mathrm{A}_{ii}^{\text{imd}} & \text{if } i = j, \\ \mathrm{A}_{ij}^{\text{conv}} & \text{otherwise.} \end{cases} \tag{3.32}$$

3. Diagonalize $\mathbf{A}^{\text{corr}}$ to obtain the intermittent eigenfrequencies and eigenvector matrix, $\mathbf{w}^{\text{int}}$ and $\mathbf{B}^{\text{int}}$ respectively. Contrary to $\mathbf{A}^{\text{imd}}$, $\mathbf{A}^{\text{corr}}$, by construction, satisfies constraint 2 (Eq. 3.30) but may not satisfy constraint 1 (Eq. 3.29).

4. Repeat steps 1-3, by setting $\mathbf{B}^{\text{guess}}$ as $\mathbf{B}^{\text{int}}$, updating the initial-guess.

We terminate IDA when one of the following occurs: (i) We obtained a solution within the desired precision,

$$\left\|\mathbf{w}^{\text{tar}} - \mathbf{w}^{\text{int}}\right\|_{\infty} \leq \epsilon, \tag{3.33}$$

where $\|\cdot\|_{\infty}$ is the infinity-norm. (ii) $\mathbf{w}^{\text{int}}$ is stuck at an undesired fixed-point, $\mathbf{w}^{\text{fixed}}$.

When IDA converges (case i), the required parameters for the optical tweezers can be calculated using the following equations,

$$\omega_i^{\text{opt}} = \sqrt{|\mathrm{A}_{ii}^{\text{corr}} - \mathrm{A}_{ii}^{\text{conv}}|}, \tag{3.34}$$

$$s_i^{\text{opt}} = \text{sign}\left(\mathrm{A}_{ii}^{\text{corr}} - \mathrm{A}_{ii}^{\text{conv}}\right), \tag{3.35}$$

where ω_i^{opt} is the required trap frequency of the i-th optical tweezer and $s_i^{\text{opt}} = \pm 1$ indicates whether the optical potential should be trapping or anti-trapping respectively.

An undesired fixed-point (case ii), $\mathbf{w}^{\text{fixed}}$, occurs when the eigenvectors of $\mathbf{A}^{\text{imd}}$ and $\mathbf{A}^{\text{corr}}$ are identical, i.e., the eigenvectors are invariant under operator T. This prevents the update of the initial-guess eigenvector matrix, $\mathbf{B}^{\text{guess}}$, in step 4, resulting in an infinite loop. We now look at the conditions required for the eigenvectors to be invariant under operator T.

Suppose the eigenvectors of $\mathbf{A}^{\text{imd}}$ and $\mathbf{A}^{\text{corr}}$ are identical,

$$\mathrm{A}_{ij}^{\text{imd}} = \sum_m \mathrm{B}_{im}^{\text{guess}} \left(\mathrm{w}_m^{\text{tar}}\right)^2 \mathrm{B}_{jm}^{\text{guess}}, \tag{3.31}$$

$$\mathrm{A}_{ij}^{\text{corr}} = \sum_m \mathrm{B}_{im}^{\text{int}} \left(\mathrm{w}_m^{\text{int}}\right)^2 \mathrm{B}_{jm}^{\text{int}}, \tag{3.36}$$

$$\mathbf{B}^{\text{fixed}} = \mathbf{B}^{\text{guess}} = \mathbf{B}^{\text{int}}. \tag{3.37}$$

Let $\delta\mathbf{A} = \mathbf{A}^{\text{corr}} - \mathbf{A}^{\text{imd}}$ and $\delta\mathbf{d} = \left(\mathbf{w}^{\text{int}}\right)^{\circ 2} - \left(\mathbf{w}^{\text{tar}}\right)^{\circ 2}$. Here, $(\,\cdot\,)^{\circ 2}$ is the Hadamard square, i.e., the element-wise square.

$$\delta\mathrm{A}_{ij} = \sum_m \mathrm{B}_{im}^{\text{fixed}} \delta\mathrm{d}_m \mathrm{B}_{jm}^{\text{fixed}}. \tag{3.38}$$

Notice, by equation 3.32, $\delta\mathbf{A}$ is a hollow symmetric matrix, i.e., its diagonal elements are zeroes.

$$\sum_m \mathrm{B}_{im}^{\text{fixed}} \delta\mathrm{d}_m \mathrm{B}_{jm}^{\text{fixed}} = 0, \quad \forall\ i = j, \tag{3.39}$$

$$\sum_m \left(\mathrm{B}_{im}^{\text{fixed}}\right)^2 \delta\mathrm{d}_m = 0, \quad \forall\ i. \tag{3.40}$$

From Eq. 3.40, a fixed point occurs only if,

$$\delta\mathbf{d} \in \text{RightNullSpace}\Big(\left(\mathbf{B}^{\text{fixed}}\right)^{\circ 2}\Big). \tag{3.41}$$

For $\left(\mathbf{B}^{\text{fixed}}\right)^{\circ 2}$ to have a non-empty right null space, it must not be full rank. Furthermore, this implies its determinant must be zero.

Differential evolution (DE)

To address the issue of terminating at fixed points of IDA, we consider utilizing the technique of differential evolution (DE). DE is an evolutionary algorithm used to perform non-linear optimization of an objective function, where the objective function need not be differentiable. It does so by maintaining an ensemble of trial parameters as a generation and specifying a protocol to modify the trial parameters and build the following generation of trial parameters. As DE iteratively generates new generations of trial parameters, a biasing in the modification protocol allows for the objective function to be optimized as DE progresses in the number of iterations.

In our particular case, DE is initialized with the parameters: an initial ensemble of $\mathbf{A}$-matrices, $\left\{\mathbf{A}^{(p)}\right\}_p$, the conventional $\mathbf{A}$-matrix, the target eigenfrequencies, $\mathbf{w}^{\text{tar}}$ and the parameters for the mutation step in DE, $\{\zeta, \kappa, \eta\}$. Intuitively, the parameter ζ is a biasing factor to the best performing member of the ensemble and parameters κ and η introduce randomness allowing the algorithm to explore a larger parameter space. These DE parameters are typically set empirically. A DE step mutates the initial ensemble of parameters in the following manner:

1. $\left\{\mathbf{A}^{(p)}\right\}_p$ are diagonalized to obtain eigenfrequencies $\left\{\mathbf{w}^{(p)}\right\}_p$.
2. Determine the index for the best performing member of the ensemble, p^*:

$$p^* = \underset{p}{\text{argmin}}\left(\left\|\mathbf{w}^{(p)} - \mathbf{w}^{\text{tar}}\right\|_\infty\right). \tag{3.42}$$

3. Define $\mathbf{x}^{(p)} = \text{diag}\left(\mathbf{A}^{(p)}\right)$. The ensemble of $\left\{\mathbf{x}^{(p)}\right\}_p$ is mutated in the following way for each index p, the mutation operator is represented as u:

$$u\left(\mathbf{x}^{(p)}; \zeta, \kappa\right) = \mathbf{x}^{(p)} + \zeta\left(\mathbf{x}^{(p^*)} - \mathbf{x}^{(p)}\right) + \kappa\left(\mathbf{x}^{(q)} - \mathbf{x}^{(r)}\right) \tag{3.43}$$

with $\{p, q, r\}$ mutually different, and q and r chosen randomly. With probability η, the mutated value of each element of $\mathbf{x}^{(p)}$ is kept, and we proceed with $\mathbf{x}'^{(p)}$ given by:

$$\mathrm{x}_i'^{(p)} = \begin{cases} \left[u\left(\mathbf{x}^{(p)}; \zeta, \kappa\right)\right]_i & \text{with probability } \eta, \\ \mathrm{x}_i^{(p)} & \text{otherwise.} \end{cases} \tag{3.44}$$

4. Using $\left\{\mathbf{x}'^{(p)}\right\}_p$, we reconstruct $\left\{\mathbf{A}'^{(p)}\right\}_p$:

$$\mathrm{A}_{ij}'^{(p)} = \begin{cases} \mathrm{x}_i'^{(p)} & \text{if } i = j, \\ \mathrm{A}_{ij}^{\mathrm{conv}} & \text{otherwise.} \end{cases} \tag{3.45}$$

After which, we diagonalize $\left\{\mathbf{A}'^{(p)}\right\}_p$ to obtain $\left\{\mathbf{w}'^{(p)}\right\}_p$ and $\left\{\mathbf{B}'^{(p)}\right\}_p$. For each index p, we only keep $\mathbf{A}'^{(p)}$ if $\mathbf{w}'^{(p)}$ is closer to $\mathbf{w}^{\mathrm{tar}}$ compared to $\mathbf{w}^{(p)}$, else we revert back to $\mathbf{A}^{(p)}$, i.e.,

$$\mathbf{A}''^{(p)} = \begin{cases} \mathbf{A}'^{(p)} & \|\mathbf{w}'^{(p)} - \mathbf{w}^{\mathrm{tar}}\|_\infty < \|\mathbf{w}^{(p)} - \mathbf{w}^{\mathrm{tar}}\|_\infty \\ \mathbf{A}^{(p)} & \text{otherwise.} \end{cases} \tag{3.46}$$

5. Finally, we output the modified ensemble, $\left\{\mathbf{A}''^{(p)}\right\}_p$, and the corresponding eigenvectors, $\left\{\mathbf{B}''^{(p)}\right\}_p$.

Iterative diagonalization algorithm with differential evolution (IDADE).

In this section, we detail the process of incorporating DE into the IDA. Henceforth, we refer to this combined algorithm as the iterative diagonalization algorithm with differential evolution (IDADE), a flowchart of which is shown in Fig. 3.4.

To initialize IDADE, we specify the following parameters: the conventional $\mathbf{A}$-matrix, $\mathbf{A}^{\mathrm{conv}}$, the target eigenfrequencies, $\mathbf{w}^{\mathrm{tar}}$, An ensemble of initial-guesses for the eigenvector matrix, indexed by population index p, $\{\mathbf{B}^{(p)}\}_p$, a desired accuracy, ϵ, DE parameters, $\{\zeta, \kappa, \eta\}$ and a maximum iteration of IDA, K. The procedure for the IDADE protocol is as follows:

1. Apply IDA in parallel for each initial-guess, until at least one instance converges, or we reach a maximum user-defined iteration number K.

2. In case of no convergence, DE is applied to the ensemble of **A**-matrices produced at the K-th iteration of IDA.

3. The output of DE is a new set of initial-guesses for the eigenvector matrix, $\{\mathbf{B}'^{(p)}\}_p$.

4. Repeat step 1-3 using the DE-produced ensemble of initial-guesses eigenvector matrix, $\{\mathbf{B}'^{(p)}\}_p$.

We terminate IDADE when we obtain a solution within a desired precision, ϵ, or we reached the user-defined maximum number of iterations of IDADE performed.

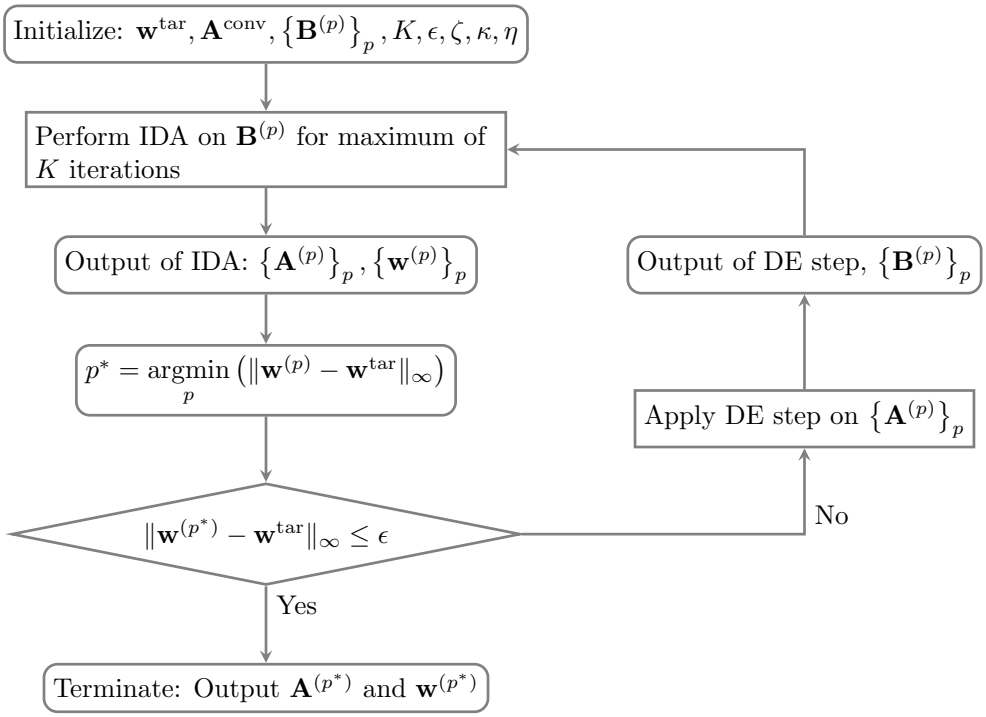


Figure 3.4: **Flowchart for iterative diagonalization algorithm with differential evolution (IDADE).**

Results for $N = 5, 10$ and 30 ions

To benchmark the performance of the IDADE protocol, we generate a dataset containing sets of phonon mode eigenfrequencies for which we know a solution exist. To generate such

a dataset, we randomly choose a set of optical tweezer parameters, $\left(\omega_i^{\text{opt}}, s_i^{\text{opt}}\right)$. Then, we calculate the **A**-matrix of the hybrid system with the optical tweezer parameters obtained. Finally, we diagonalize the **A**-matrix to obtain the phonon mode eigenfrequencies, **w**. We store this set of phonon mode eigenfrequencies in our dataset to use as our target set of phonon mode eigenfrequencies when testing the IDADE protocol.

We tested the protocol for programming the phonon mode eigenfrequencies of $N = 5, 10$ and 30 systems using a data set of 1000 points. As the experimental limit of mode stabilization is of order $2\pi \times 10$ Hz [36], this value sets the desired accuracy, ϵ, for the IDADE protocol. Other parameters of the IDADE protocol include the values for the DE parameters, $\zeta = 0.9$, $\kappa = 0.5$, $\eta = 0.5$, and an ensemble of 100, randomly set, initial-guess eigenvectors. Empirically, we found that for $N = 5, 10$ ions, the required number of IDA iteration is $K = 100$, and for $N = 30$, $K = 600$.

Fig. 3.5 demonstrates that, with the above parameters, the IDADE protocol was capable of programming a wide range of target spectra within the desired tolerance. It also depicts a decrease in the success probability with increasing system size. This effect can potentially be mitigated by further optimization of the IDADE parameters.

Additionally, we observed a significant increase in accuracy, to 10^{-8} Hz, when the target mode frequencies are much larger than the conventional mode frequencies. This is due to a decoupling in the phonon modes when the Coulomb potential is negligible compared to the sum of the optical and conventional trapping potentials. Consequently, each ion acts as an independent oscillator, wherein the oscillation frequency can be individually set.

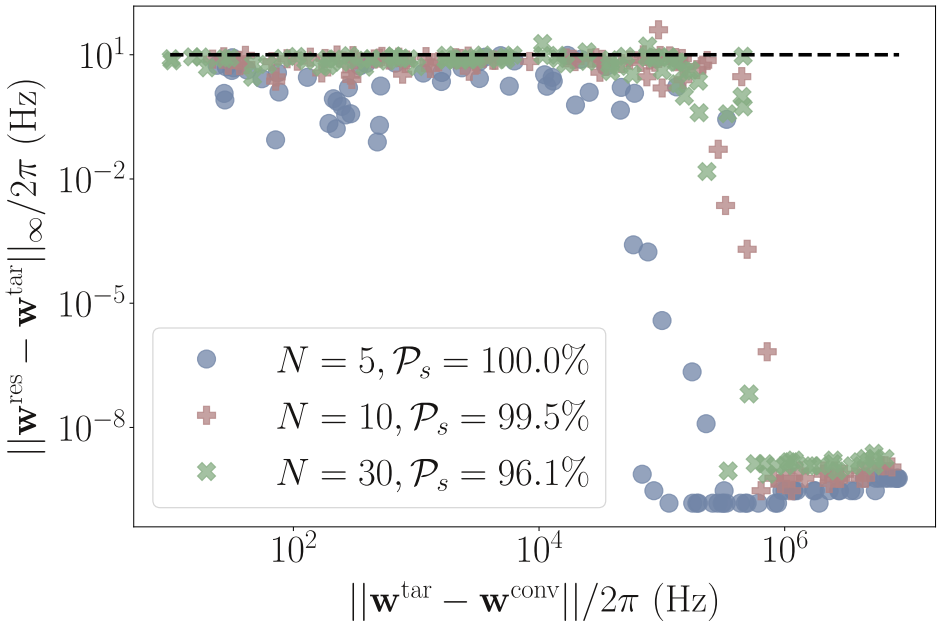


Figure 3.5: **Performance of the IDADE protocol.** Here, the IDADE protocol is benchmarked for $N = 5, 10$ and 30 ion-systems using 1000 randomly generated target phonon mode frequencies (only 100 are shown for simplicity). The error in the normal mode eigenfrequecies, $\|\mathbf{w}^{\mathrm{res}} - \mathbf{w}^{\mathrm{tar}}\|_\infty$, is compared with the desired accuracy, $\epsilon/2\pi = 10$ Hz (dashed black line). All the points below the dashed line are considered 'successful' and we show the success probability, $\mathcal{P}_s$, of the protocol for the different number of ions.

3.4.2 Eigenvector control

Statement of problem

A statement of the phonon mode eigenvector control problem is as follows:

- Determine the required parameters on the optical tweezers $\left(\omega_i^{\mathrm{opt}}, s_i^{\mathrm{opt}}\right)$ such that the trapped-ion system has a target set of phonon mode eigenvectors, $\mathbf{B}^{\mathrm{tar}}$.

Or equivalently, the problem could be mathematically formulated as such:

- Determine an $\mathbf{A}$-matrix that satisfies the following constraints simultaneously:

$$\mathrm{A}_{ij} = \sum_m \mathrm{B}^{\mathrm{tar}}_{im}\mathrm{D}_{mn}\mathrm{B}^{\mathrm{tar}}_{jn}, \text{ where } \mathbf{D} \text{ is an arbitrary diagonal matrix.} \tag{3.47}$$

$$\mathrm{A}_{ij} = \mathrm{A}^{\mathrm{conv}}_{ij}, \ \forall\ i \neq j. \tag{3.30}$$

  Here, constraint 1 (Eq. 3.47) ensures that the $\mathbf{A}$-matrix has the target eigenvectors, $\mathbf{B}^{\mathrm{tar}}$, whereas, constraint 2 (Eq. 3.30), is identical to in the previous section.

Similar to the previous problem of eigenfrequencies control, the eigenvector control problem has the following properties: (1) a solution is not guaranteed to exist and (2) solutions may not be unique. These properties are derived in Appendix B.

Eigenvector control protocol

We can rewrite Eq. 3.47 as the following:

$$\sum_{i,j} \mathrm{B}^{\mathrm{tar}}_{im}\mathrm{A}_{ij}\mathrm{B}^{\mathrm{tar}}_{jn} = \mathrm{D}_{mn} = \begin{cases} 0 & \text{if } m \neq n, \\ \text{arbitrary} & \text{otherwise.} \end{cases} \tag{3.48}$$

By defining $\mathrm{C}_{ijmn} = \mathrm{B}^{\mathrm{tar}}_{im}\mathrm{B}^{\mathrm{tar}}_{jn}$, Eq. 3.48 can be rewritten as:

$$\sum_{i,j} \mathrm{C}_{ijmn}\mathrm{A}_{ij} = 0, \quad \forall\ m \neq n, \tag{3.49}$$

Notice, Eq. 3.49 defines a linear problem. We are able to solve Eq. 3.49 using singular value decomposition (SVD) and determine the right null space of $\mathbf{C}$. Let the right null space of $\mathbf{C}$ be $\left\{\hat{\mathbf{A}}^{(k)}\right\}_k$, this defines a basis of matrices with the target eigenvector matrix $\mathbf{B}^{\mathrm{tar}}$.

To determine an $\mathbf{A}$-matrix that satisfies the off-diagonal contraints (Eq. 3.30), we can consider solving for coefficients $\{\gamma_k\}_k$ in the following linear problem:

$$\sum_k \hat{\mathrm{A}}^{(k)}_{ij}\gamma_k = \mathrm{A}^{\mathrm{conv}}_{ij}, \quad \forall\ i \neq j. \tag{3.50}$$

Eq. 3.50 can be solved using QR decomposition and back substitution.

Similar to in the eigenfrequencies control problem , we can use Eqs. 3.34 and 3.35 to obtain the required trap frequency and sign of the i-th optical tweezer, ω_i^{opt} and s_i^{opt}, respectively.

3.5 Enhancing phonon mode coupling in a mass-imbalanced system

In recent years, extensive research has been conducted on multi-species trapped-ion systems to investigate the benefits they offer for scaling up quantum processors [37]. Another key advantage of the multi-species system is the ability to perform sympathetic cooling. This technique enables the simultaneous cooling of the motional modes of the system during QIP experiments, mitigating errors due to heating effects. However, it has been shown that when there exist a significant mass-imbalance in the system, there is an effective decoupling between the ions. Consequently, this introduces additional errors in the quantum computation and restricts the efficiency of sympathetic cooling [38].

In this section, we demonstrate that the hybrid trapped-ion system proposed in this chapter is able to mitigate the effects of mass-imbalance in a multi-species system.

3.5.1 Conventional multi-species phonon modes

Consider a conventional trapped-ion system composed of four $^{133}Ba^{+}$ ions and a $^{171}Yb^{+}$ ion at the left edge of the chain (shown in Fig. 3.6). The mass-imbalance of this system is $M_{Ba}/M_{Yb} = 0.778$ and the conventional trap strength on the ions are given in Table 3.2. The conventional trap strengths on $^{133}Ba^{+}$ is calculated from the conventional trap strengths of $^{171}Yb^{+}$ using Eqs. 1.4 to 1.6.

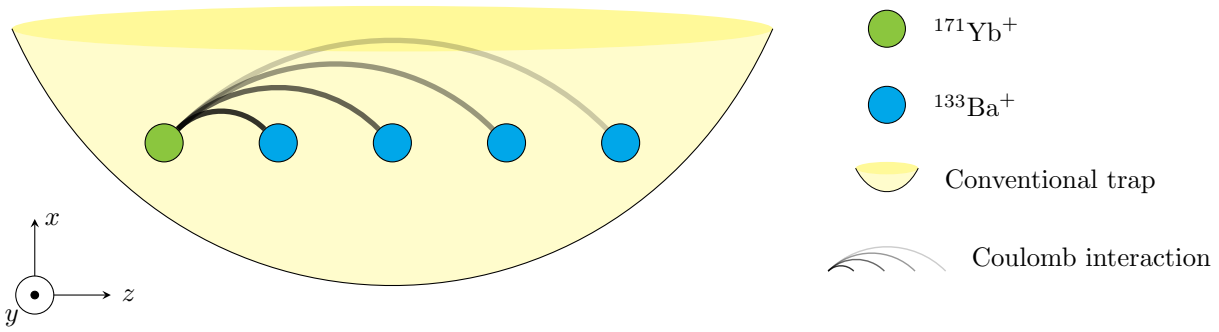


Figure 3.6: **Schematic of the potential in a multi-species conventional trapped-ion system.** Here, a single ^{171}Yb$^+$ (green node) ion, on the left edge of the chain, and four ^{133}Ba$^+$ (blue node) ions are trapped in a conventional radio-frequency trap. Additionally, the ions interact via the Coulomb interaction, for cleanliness, only the coupling between the ^{171}Yb$^+$ ion and ^{133}Ba$^+$ ions are shown, the coupling between ^{133}Ba$^+$ ions are omitted in the diagram.

	^{171}Yb$^+$	^{133}Ba$^+$
$\omega_x/2\pi$ (MHz)	1.2	1.519
$\omega_y/2\pi$ (MHz)	1.0	1.32
$\omega_z/2\pi$ (MHz)	0.2	0.227

Table 3.2: **Table of conventional trapping strengths for ^{171}Yb$^+$ and ^{133}Ba$^+$.** Eqs. 1.4 to 1.6 is used to calculate the trap strengths on ^{133}Ba$^+$ from the trap strengths of ^{171}Yb$^+$ using

Notice the transverse trapping strength of ^{133}Ba$^+$ is significantly higher than that of ^{171}Yb$^+$ in Table 3.2. This discrepancy in the transverse trapping strength leads to a decoupling effect in the transverse phonon modes of the system. This decoupling is explicitly shown in Table 3.3, where we observe a localized phonon mode for the ^{171}Yb$^+$ ion and the remaining four modes are collective phonon modes for the ^{133}Ba$^+$ ions only. A similar decoupling of phonon modes was previously observed in Section 3.3 in the case of high optical trap strengths.

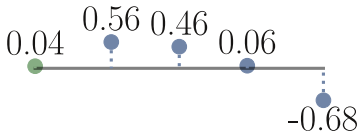
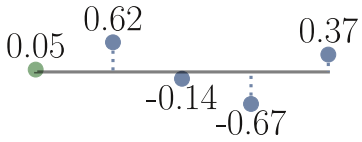
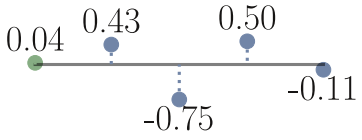
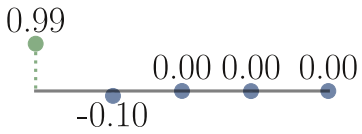
m	$\bar{B}_{im}^{(y)}$	$w_m/2\pi$ (MHz)
1		1.314
2		1.283
3		1.241
4		1.193
5		0.966

Table 3.3: **Transverse phonon modes of a multi-species trapped-ion system.** The y-direction phonon modes, eigenvectors and eigenfrequencies, of the multi-species conventional trapped ion system described in Fig. 3.6 with trapping parameters given in Table 3.2. The numbers in the eigenvector plots correspond to normalized vectors that quantify the relative participation of each ion in the phonon modes. Notice, the $^{171}Yb^{+}$ ion barely participates in the first to fourth mode and the participation of the $^{133}Ba^{+}$ ions in fifth mode is small.

3.5.2 Enhancing transverse modes

Consider now introducing an optical tweezer on the $^{171}Yb^{+}$ ion to compensate for the discrepancy in the y-trap frequency, the system is shown in Fig. 3.7. The required optical

trap strength can be derived from Eq. 3.28,

$$\left(\omega_y^{\text{opt}}\right) = \sqrt{\left(\omega_y^{\text{Ba}}\right)^2 - \left(\omega_y^{\text{Yb}}\right)^2}. \tag{3.51}$$

For the conventional trap strengths given in Table 3.2, the optical y-trap strength required is $\omega_y^{\text{opt}}/2\pi = 0.861$ MHz.

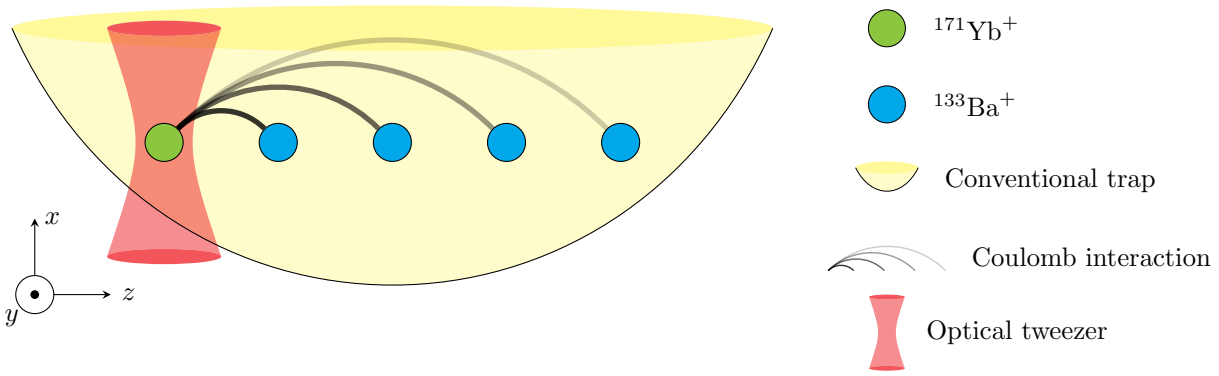


Figure 3.7: **Schematic for optical tweezer enhanced phonon mode coupling in a multi-species trapped-ion system.** This schematic describes the system in Fig. 3.6 with an additional optical tweezer applied onto the ^{171}Yb$^+$ ion. By appropriately adjusting the optical trap parameters, given by Eq. 3.51, we are able to recover the phonon mode coupling between the ^{171}Yb$^+$ and ^{133}Ba$^+$ ions as shown in Table 3.4.

Notice in Table 3.4, the inter-species coupling of the y-phonon modes of the system has been restored by the introduction of the optical tweezer on the ^{171}Yb$^+$ ion. The new phonon modes of the multi-species system resemble the phonon modes of a single species system in a conventional trap (Table 1.1).

However, as Eq. 1.8 implies that $\bar{\mathbf{B}}^{(\alpha)}$ is not unitary for a multi-species system. It is not possible to fully restore the modes of a conventional single-species system in a multi-species setup.

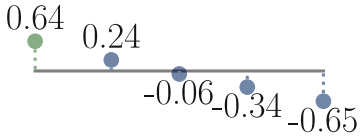
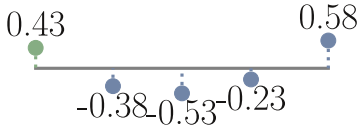
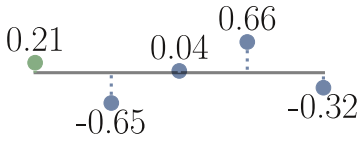
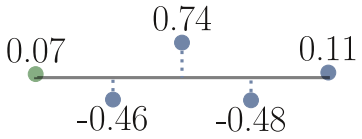
m	$\bar{B}_{im}^{(y)}$	$w_m/2\pi$ (MHz)
1		1.32
2		1.302
3		1.275
4		1.237
5		1.192

Table 3.4: **Optical tweezer enhanced transverse phonon modes of a multi-species trapped-ion system.** The y-direction phonon modes, eigenvectors and eigenfrequencies, of the multi-species trapped ion system described in Fig. 3.7. The optical tweezers on the $^{171}Yb^+$ ion has a y-trap frequency of $\omega_y^{\text{opt}}/2\pi = 0.861$ MHz. The numbers in the eigenvector plots correspond to normalized vectors that quantify the relative participation of each ion in the phonon modes. Notice, compared to Table 3.3, the participation of the $^{171}Yb^+$ ion and $^{133}Ba^+$ ions in all modes are significant.

3.6 Conclusion

In this chapter, we studied the effects of introducing optical tweezers into the conventional trapped-ion system. We proposed a hybrid trapped-ion system, where the conventional trapped-ion system is augmented with controllable optical tweezers focused on each ion. This permits previously unattainable degrees of control over the phonon modes of the

trapped-ion system. Furthermore, we introduced the protocols developed to program this hybrid trapped-ion system for desired phonon mode structure. Finally, we investigated an appealing application of this hybrid-trapped-ion system to mitigate the effects of motional decoupling due to mass-imbalance in a multi-species trapped-ion system.

Chapter 4

State readout of a trapped-ion system using machine learning techniques

In all QIP experiments, the ability to measure the state at the end of the experiment is critical to understanding the results. For the trapped-ion platform, this measurement is performed using fluorescent measurement, where resonant laser beams are used to couple one of the qubit state to an excited state with an optical transition wavelength. This induces a closed cycling transition where ions in this state continually fluoresce, i.e., emit photons, distinguishing them from the ions, in the other state, that do not. In a multi-ion setup, we require the ability to discriminate between the photons emitted from each ion, therefore, a sensor with spatial resolution, e.g. multi-channel photo-multiplier tube (MC-PMT) or a charge-coupled device (CCD), is utilized to capture the photons. Thresholding methods are then used on the image (photon counts) to identify the state of the ions.

In recent years, machine learning methods have been investigated to assist in the process of identifying the state of the ions from the image [39]. They have been found to be just as accurate as, if not more accurate than, traditional approaches.

In this chapter, we study the mechanism behind the fluorescent measurement of the qubit states of trapped-ion systems using a detector with spatial resolution. We developed a feedforward neural network (FFNN) that is capable of learning the point spread function (PSF) of the ions in a multi-ion system and determine the state of the ion. We observe that the neural network has the form of a linear classifier, not unlike traditional adaptive threshold methods , where the weights and biases of the FFNN specify the contribution of

each pixel and the threshold for the fluorescent measurement respectively. Furthermore, to improve the generality and scalability of the machine learning algorithm for state readout, we developed an algorithm to determine a region of interest for each ion using a calibration image generated during the cooling phase of the experiment. The FFNN is then applied only on the region of interest for each ion. This form a scalable algorithm for the state readout of trapped-ion systems.

4.1 Detection theory

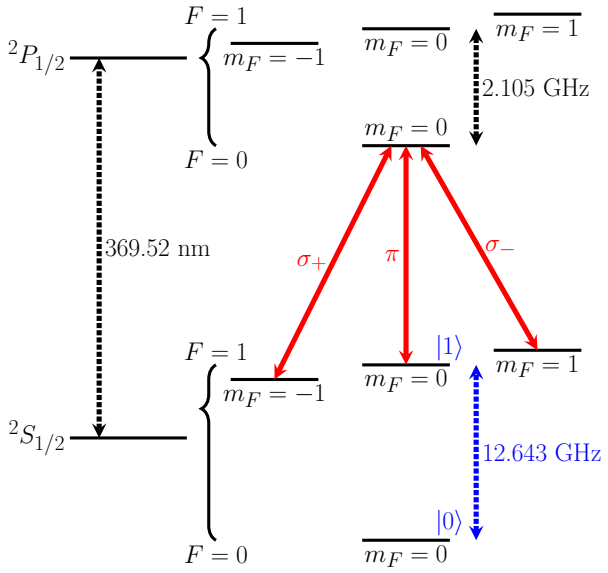


Figure 4.1: **Relevant energy levels in the fluorescent measurement of $^{171}Yb^+$ ions.** During measurement, resonant lasers, of $\sigma_\pm$ and π polarities, are used to drive the transitions (red) coupling the $|S_{1/2}, F = 1\rangle$ submanifold of states to the $|P_{1/2}, F = 0, m_F = 0\rangle$ state. Notice, the above mentioned lasers only off-resonantly drive the transition from $|S_{1/2}, F = 0, m_F = 0\rangle$ state to $|P_{1/2}, F = 1\rangle$ submanifold of states, with detuning equal to the sum of the hyperfine splittings of $S_{1/2}$ and $P_{1/2}$.

In this section, we give an overview of detection theory. Fig. 4.1 shows the energy levels that are used in the fluorescent measurement of $^{171}Yb^+$. As mentioned in Section 1.1, the hyperfine $S_{1/2}$ states are used as qubit states for $^{171}Yb^+$. By applying lasers resonant with the

transition (detection beam) from the $|1\rangle$ state to the $\left|P_{1/2}, F=0, m_F=0\right\rangle$ state, ions in the $|1\rangle$ state undergo a cycling transition and fluoresce. Whereas, for ions in the $|0\rangle$ state, the detection beam is detuned from their allowed transition to the $\left|P_{1/2}, F=1, m_F=0, \pm 1\right\rangle$ states by a factor of $\omega_{\mathrm{HFS}} + \omega_{\mathrm{HFP}}$, where ω_{HFS} and ω_{HFP} are the hyperfine splittings of the $S_{1/2}$ and $P_{1/2}$ state respectively. Therefore, the $|0\rangle$ and $|1\rangle$ states are distinguishable.

Notice, additional resonant laser beams, of different polarities, $\sigma_{\pm}$ and π, are used to couple the $\left|S_{1/2}, F=1, m_F=\pm 1\right\rangle$ states to the $\left|P_{1/2}, F=0, m_F=0\right\rangle$. This is done to ensure that ions in the $\left|P_{1/2}, F=0, m_F=0\right\rangle$ state decay back to the $\left|S_{1/2}, F=1\right\rangle$ states and the cycling transition is closed.

The photon statistics for the bright ion, i.e., ions in state $|1\rangle$ that fluoresce under the detection beam, would follow a Poissonian distribution dependent on the following parameters: The radiative linewidth, γ, the photon collection efficiency of the detector, η, the detuning of the detection beam from the cycling transition, δ, the intensity of the detection beam, I_{D}, and the detection time, τ_{D}. The photon count probability distribution has the following form,

$$\Pr(X = k; |1\rangle) = \mathrm{Poisson}(k; \lambda_0) = \frac{\lambda_0^k e^{-\lambda_0}}{k!}, \tag{4.1}$$

$$\lambda_0 = \tau_{\mathrm{D}} \eta \frac{s\frac{\gamma}{2}}{1 + s + \left(\frac{2\delta}{\gamma}\right)^2}. \tag{4.2}$$

Here, X is the number of photons emitted by the ion, λ_0 is the mean photon count, $s = I_{\mathrm{D}}/I_{\mathrm{sat}}$ is the saturation parameter for the detection beam, I_{D} is the intensity of the detection beam and I_{sat} is the saturation intensity.

The above distribution of photon count assumes that there is no leakage between the $|1\rangle$ and $|0\rangle$ states. To account for leakage due to the off-resonant excitations, the photon count probability distribution for the dark and bright state is modified by convolving the aforementioned Poissonian distribution with an exponential distribution (probability

distribution corressponding to state leakage) [40],

$$\Pr(X=k;|0\rangle) = \exp\left(-\frac{\alpha_1\lambda_0}{\eta}\right)\left[\delta_n + \frac{\frac{\alpha_1}{\eta}}{\left(1-\frac{\alpha_1}{\eta}\right)^{k+1}}\Gamma\left(k+1,\left(1-\frac{\alpha_1}{\eta}\right)\lambda_0\right)\right], \tag{4.3}$$

$$\Pr(X=k;|1\rangle) = \frac{\lambda_0^k \exp\left(-\left(1+\frac{\alpha_2}{\eta}\right)\lambda_0\right)}{k!} + \frac{\frac{\alpha_2}{\eta}}{\left(1+\frac{\alpha_2}{\eta}\right)^{k+1}}\Gamma\left(k+1,\left(1+\frac{\alpha_2}{\eta}\right)\lambda_0\right), \tag{4.4}$$

$$\Gamma(a,b) = \int_0^b t^{a-1}e^{-t}\,dt. \tag{4.5}$$

Here, α_1 and α_2 are the leakage probabilities, dependent on the atomic parameters, Γ is the lower incomplete gamma function and δ_n is the Kronecker delta. An example of the photon count distribution for $^{171}\mathrm{Yb}^+$ is shown in Fig. 4.2. The leakage probabilities for $^{171}\mathrm{Yb}^+$ are $\alpha_1 = 1.686\times10^{-7}$ and $\alpha_2 = 8.264\times10^{-6}$.

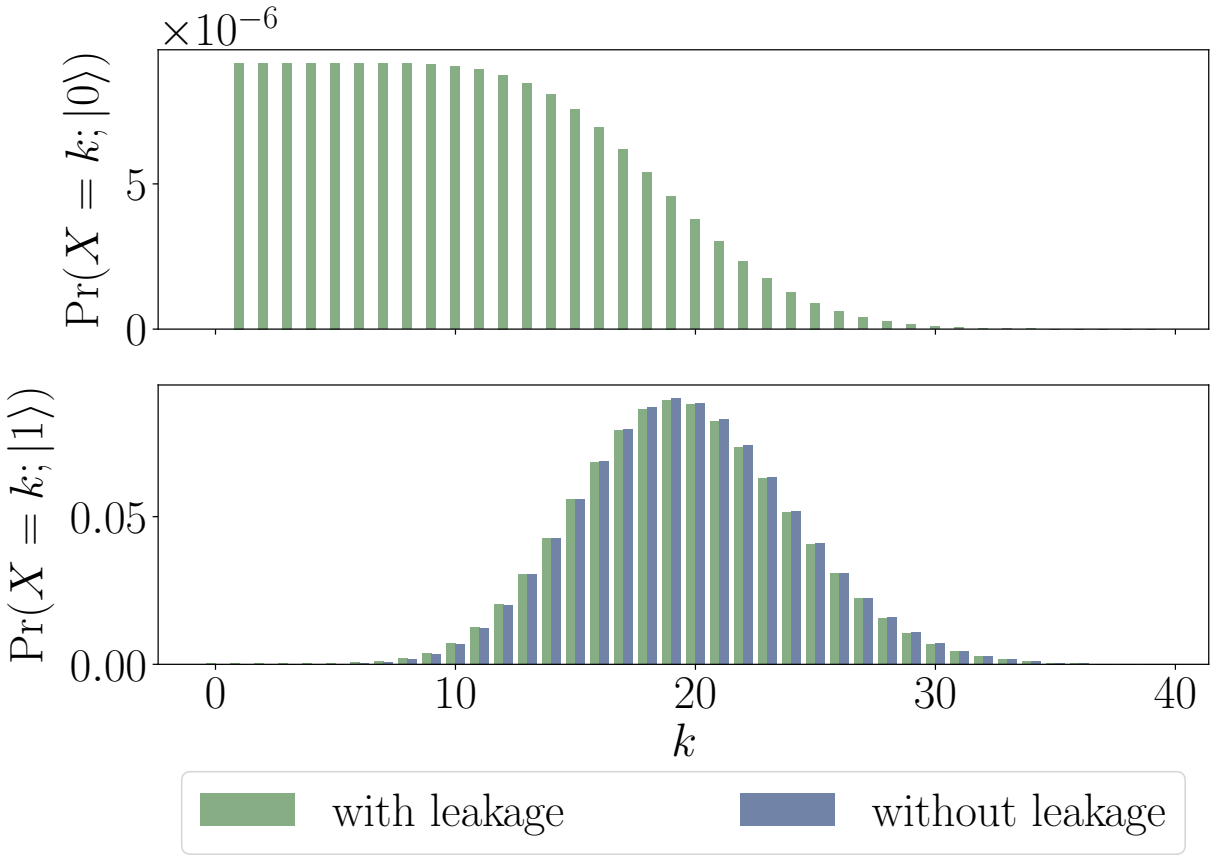


Figure 4.2: **Photon count statistics for bright and dark $^{171}Yb^{+}$ ions.** Here, the mean bright photon count, $\lambda_0 = 19.67$. The probability for $X = 0$ photons is omitted in the plot for the dark state, $|0\rangle$. Without leakage, we do not obtain any photons for state $|0\rangle$.

To quantify the performance of a protocol for state readout, we define a fidelity, $\mathcal{F}$, for the process using the success probability, i.e., probability the protocol correctly classifies the state of the ion. Based on the above distributions, we obtain a theoretical maximal fidelity, $\mathcal{F}_{\max}$, for state readout of,

$$\mathcal{F}_{\max} = \frac{1}{2}\sum_k \max\left(\Pr(X = k; |0\rangle), \Pr(X = k; |1\rangle)\right). \tag{4.6}$$

For the distribution in Fig. 4.2, $\mathcal{F}_{\max} = 0.99969$.

In the above we have specified the distribution for the photon counts emitted by the dark and bright state. However, for a multi-ion system, a detector with spatial resolution is required, as such, we have to determine the spatial distribution of the photons. Assuming the ions are point sources of photons, the ions have a point spread function (PSF) on the detector that describes the probability that a photon emitted by the ion impacts the detector at a particular position. In the case where the imaging system is diffraction limited, we obtain a PSF that follows the Airy pattern,

$$\mathrm{PSF}(x, y) = \frac{2J_1\left(ka \sin[\theta(x, y)]\right)}{ka \sin[\theta(x, y)]}. \tag{4.7}$$

Here, J_1 is the first-order Bessel function of the first kind, k is the wave number of the incident light, a is the radius of the aperture, θ is the angle of observation and (x, y) is the coordinate on the detector.

As detectors possess finite spatial resolution, they produce pixelized images. To account for this, the point spread function has to be discretized, the result is the discretized point spread function (DPSF).

$$\mathrm{DPSF}_{mn} = \int_{y_n}^{y_{n+1}} \int_{x_m}^{x_{m+1}} \mathrm{PSF}(x, y)\, dxdy, \tag{4.8}$$

$$\alpha_m = m\Delta\alpha, \text{ for } \alpha \in \{x, y\}. \tag{4.9}$$

In practice, we compute the integral using Monte-Carlo integration.

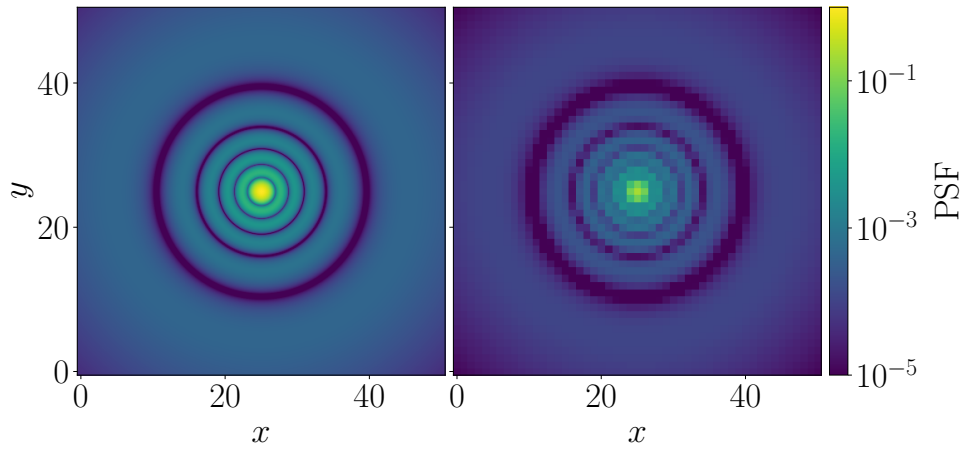


Figure 4.3: **Point spread function (PSF) of an ion .** Theoretical PSF of an ion alongside the simulated DPSF on the detector of the ion. Here, the Airy radius is 1.95 px.

In experiments, there are errors that enter in the fluorescent detection process. For the photon count distribution, noise sources include, stray photons from the background impacting the detector or dark current in the detector. We model these additional noises by a Poisson distribution for each pixel on the detector.

$$\Pr_{mn}^{\text{noise}}(X = k) = \text{Poisson}(k; \lambda_{\text{noise}}), \tag{4.10}$$

$$\lambda_{\text{noise}} = \tau_{\text{D}} R_{\text{noise}}. \tag{4.11}$$

Here, R_{noise} is the rate of erroneous photons registered by the detector. Likewise, there are potential errors in the PSF due to aberrations in the imaging system. However, these errors could be corrected via the fine-tuning of the imaging system or by post-processing methods [41].

The probability distributions supplied in this section are used to generate simulated fluorescent measurement data for the machine learning efforts that follow.

4.2 Feedforward neural network for state readout

The state readout problem is as follows: Given the image generated on the detector after the measurement procedure, determine the state of the ions in the trapped-ion system that corresponds to the image. In this section, we investigate the use of a feedforward neural network (FFNN) for state readout of a trapped-ion system. Further study determined that the neural network employs a linear classifier, similar to traditional adaptive threshold schemes, to perform state discrimination. The threshold and the contribution of each pixel in the determination of the state of an ion is learnt through the training of the neural network.

4.2.1 Neural network structure

The structure of the neural network (shown in Fig. 4.4) is as follows,

- The first layer is the input layer which takes the image produced by the detector during the fluorescent measurement, henceforth called the measurement image, X_{mn}. The number of units in this first layer is given by the dimensions of detector, $D_1 \times D_2$.
- The input layer is fully connected to the hidden layer with N (number of ions) linear units without biases.

- The hidden layer is fully connected to the output layer with N sigmoid units. The form of the sigmoid unit is shown in Table 1.2. This output is interpreted as the predicted probabilities that each ion is bright, p_k^{pred}.

- The binarization of the output layer, with a threshold of 0.5, is then used to specify the neural network predicted state, ψ_i^{pred}. This process is not applied during the training of the neural network.

The trainable parameters of the neural network include the weights between the input and hidden layer and between the hidden and output layer, $W_{mnj}^{(1)}$ and $W_{ji}^{(2)}$ respectively, and the bias of the output layer, $b_i^{(2)}$.

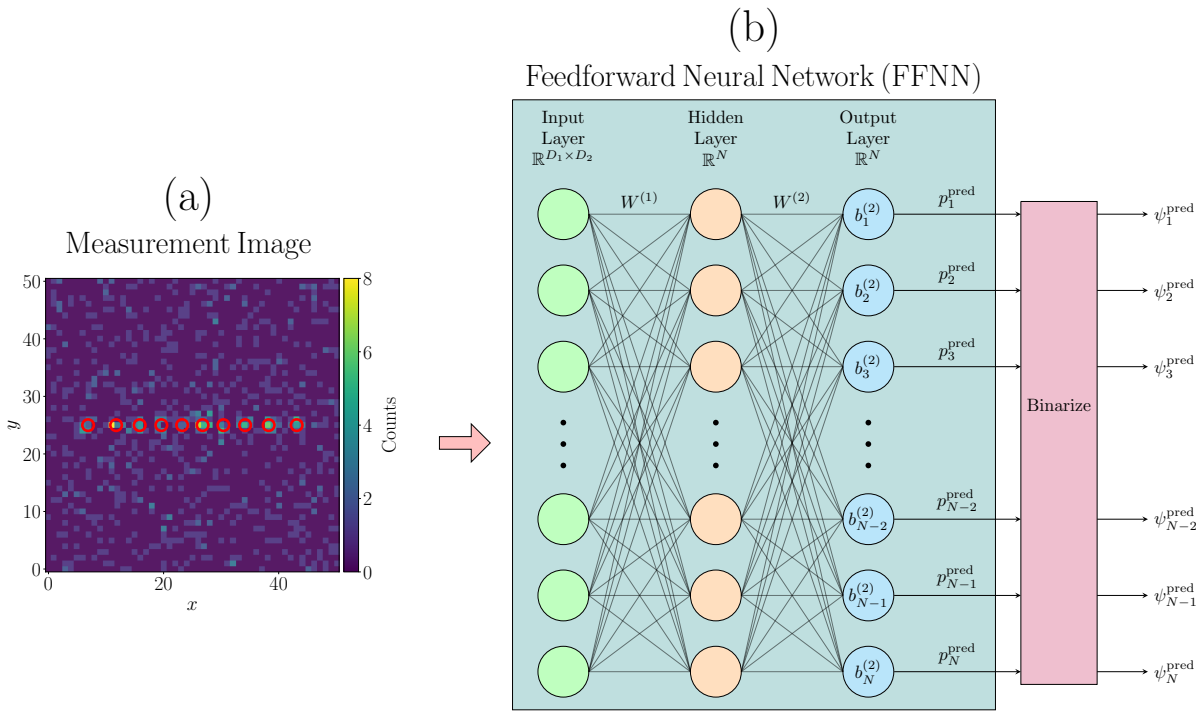


Figure 4.4: **Feedforward neural network (FFNN) for state readout.** The input of the neural network is the (a) measurement image, which is two-dimensional. In the (b) neural network structure, the input layer is displayed in its flattened form. The output of the neural network is the predicted bright probability of each ion, p_i^{pred}. By binarizing the predicted bright probability of each ion, we obtain the neural network predicted state of each ion, ψ_i^{pred}. The full details of the neural network are specified in the text.

4.2.2 Training parameters

The neural network parameters (the weights and biases) are trained by comparing the predicted bright state probabilities the neural network outputs, p_i^{pred}, and the actual state of the ion, ψ_i. Notice, as there are only two possible states, $\psi_i = 0$ or 1, the probability of the state being bright is equivalent to the state labels.

We utilized the ADAM algorithm [14] to optimize the binary cross-entropy cost function,

$$C(p_i^{\text{pred}}, \psi_i) = -\frac{1}{N} \sum_k p_i^{\text{pred}} \log(\psi_i) + (1 - p_i^{\text{pred}}) \log(1 - \psi_i) \,. \tag{4.12}$$

Other empirically chosen hyperparameters include the sizes of the training and validation data set, 10000 and 1000 respectively, a batch size of 100, a learning rate of 10^{-3} and we trained the neural networks for 100 epochs.

4.2.3 Condensed neural network

A property of the introduced FFNN is that it can be condensed, this originates from the linearity of the hidden layer. Consider the dependence of the output of the neural network on the inputs,

$$p_i^{\text{pred}} = \text{Sigmoid}\left(\sum_{m,n,j} W_{jk}^{(2)} W_{mnj}^{(1)} X_{mn} + b_i^{(2)} \right). \tag{4.13}$$

By defining condensed weights, $W_{mni}^{\text{cond}} = \sum_j W_{ji}^{(2)} W_{mnj}^{(1)}$, we obtain,

$$p_i^{\text{pred}} = \text{Sigmoid}\left(\sum_{m,n} W_{mni}^{\text{cond}} X_{mn} + b_i^{(2)} \right). \tag{4.14}$$

This corresponds to removing the hidden layer in the neural network and using the condensed weights, W^{cond}, to connect the input and output layer. Despite the fact that the neural network may be condensed, the hidden layer is needed for efficient training as it expands the parameter space.

The condensed neural network employs a linear classifier to perform state discrimination. Linear classifiers are typical protocols used for state readout. One such linear classifier is

the adaptive threshold method (ATM) with general form,

$$\psi_i^{\text{ATM}} = \text{Heaviside}\left(\sum_{m,n} W_{mni}^{\text{threshold}} X_{mn} - b_i^{\text{threshold}}\right). \tag{4.15}$$

Here Heaviside is the Heaviside step function, $W_{mni}^{\text{threshold}}$, $b_i^{\text{threshold}}$ are the weights and thresholds used in determining the state of ion-i.

There are stark similarities between the output of the condensed neural network and the adaptive threshold method. The sigmoid function in the condensed neural network output is a smooth approximation of the Heaviside step function. Further, the condensed weights and output biases in the condensed neural network are analogous to the weights and thresholds in the adaptive threshold method.

4.2.4 Results and analysis

Single ion system

We consider a simple trapped-ion system with a single trapped ytterbium ion, $^{171}\text{Yb}^{+}$. We simulated fluorescent measurement in the system with and without the noise factor in the photon count distribution. The parameters we used for the simulation include a mean photon count emitted by bright ions, $\lambda_0 = 19.67$, an Airy radius of 1.95 px for the PSF of the ion and a mean erroneous photon count per pixel, $\lambda_{\text{noise}} = 0.24$. We train the proposed FFNN on these data sets.

For the simple case of a single ion in the absence of noise, we are able to determine a maximal fidelity threshold (MFT) method, given by Eq. 4.6, and map it to the condensed neural network.

	Cost, C	Fidelity, $\mathcal{F}$
MFT without noise	0.347(12)	0.9997(5)
FFNN without noise	0.003(4)	0.9993(8)
FFNN with noise	0.097(24)	0.9764(48)

Table 4.1: **Single ion state readout fidelity of the state readout FFNN.** The performance of the MFT method without noise and the state readout FFNN in the presence and absence of noise for a trapped-ion system consisting of a single ion. Notice, without error, the infidelity, $1 - \mathcal{F}$, of the state readout FFNN and MFT method is of the same order of magnitude. In the presence of noise, as expected, the cost function and infidelity of the state readout FFNN both increase.

In Table 4.1, we observe a discrepancy between the fidelity of the FFNN without noise and MFT. FFNN without noise performs slightly worse than MFT in terms of fidelity, however, MFT has a much higher cost compared to FFNN without noise. As the neural network optimizes the cost function, the difference in the form of the fidelity and cost function results in the slight discrepancy in fidelity.

In Fig. 4.5, we show that the neural network learns the characteristics of the PSF in its weights. There are striking similarities in the structure of the condensed weights, W^{cond}, of the FFNN without noise and the DPSF of the ion. They both exhibit alternating bright and dark fringes surrounding a centroid with the same coordinates.

In contrast, the FFNN trained with error has only learnt the centroid of the DPSF. This is logically sound as the presence of noise implies that photons beyond the centroid of the DPSF almost certainly originated from the noise sources. Based on this, the neural network, in the presence of noise, determines a subset of important pixels, i.e., region of interest (ROI), for the state readout of an ion. For the parameters we considered, we observe the neural network use a 3 by 3 pixel region as the ROI for an ion.

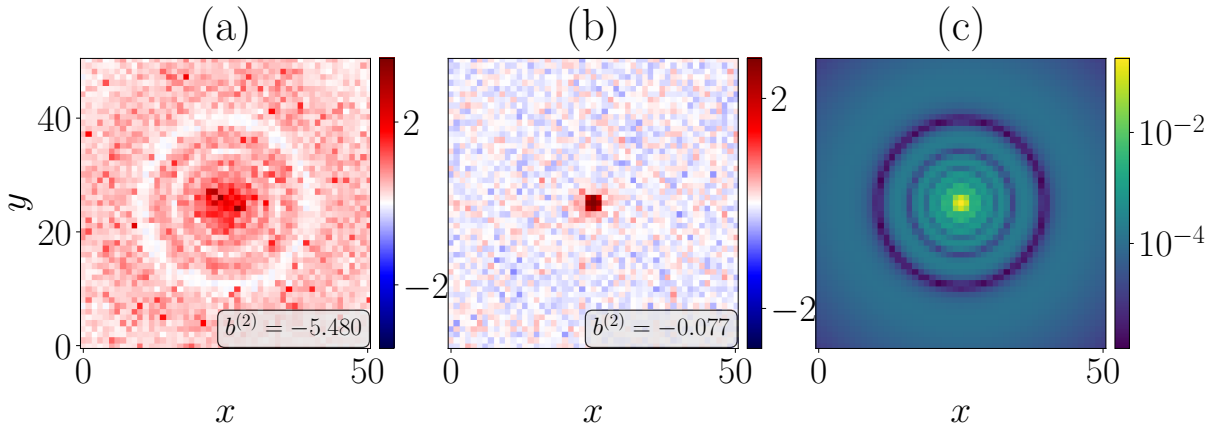


Figure 4.5: **Neural network learnt discrete point spread function (DPSF).** Condensed weights and bias, W^{cond} and $b^{(2)}$, for the FFNN trained (a) without noise and (b) with noise. (c) Simulated DPSF on the detector of the ion. Notice the stark similarities between the condensed weights of the neural networks and the DPSF of the ion. This implies that the state readout FFNN learns the form of the DPSF of the ion during training. (a) Without noise, the state readout FFNN learns the full structure of the DPSF, (b) with noise, the state readout FFNN learns only the centroid of the DPSF.

Multi-ion system

Next we consider a trapped-ion system with multiple ions. The PSF for the ions are assumed to be identical, and is specified by the Airy pattern (Eq. 4.7). One thing to note is that the DPSF depends on the relative position of the center of the Airy disk in the center pixel. This implies the DPSF is only translationally invariant in periods of a single pixel. This means that even though we assume the PSF of the ions are the same, their DPSF on the detector can be more than just translationally different.

In Fig. 4.6, we trained the FFNN for the state readout of 2 ions and compared the condensed weights with the PSF of the ions. The FFNN, as shown previously, demonstrates the ability to learn the centroid of the PSF for each ion and utilizes it to weight the pixels before applying a threshold for state readout.

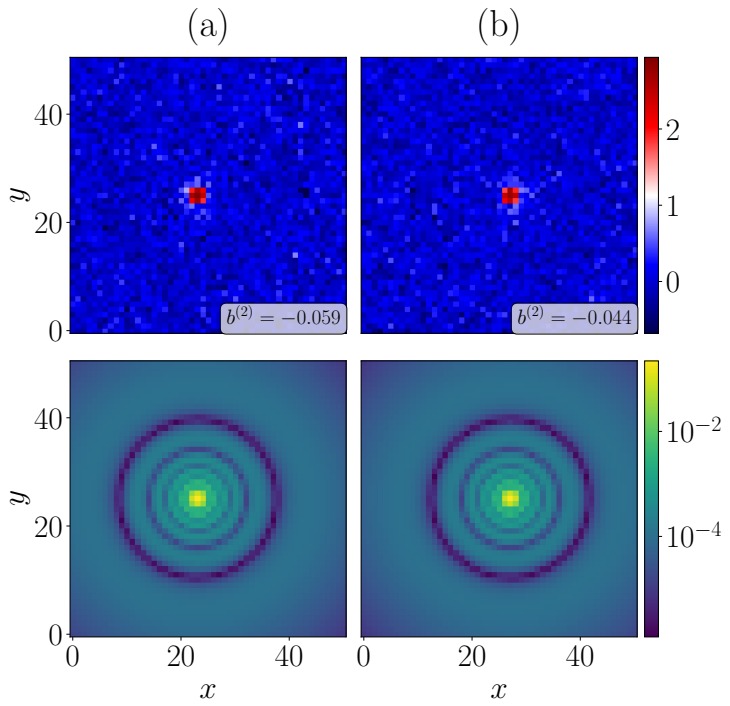


Figure 4.6: **Neural network learnt region of interest (ROI) for two ions.** (Top) Condensed weights and bias, W^{cond} and $b^{(2)}$, for the FFNN and (bottom) simulated DPSF on the detector for (a) ion 1 and (b) ion 2. The 2 ions are separated by 4 px on the detector. Notice, the condensed weights is similar to the centroid of the DPSF of the ions, this implies the state readout FFNN learns the centroid of the DPSF of each ion during training.

In the case of a 2 ions system, there is also a cross-talk error in addition to the noise sources previously mentioned. Cross-talk depends on the overlap between the DPSF of the two ions which grows as the separation between the ions is decreased. Fig. 4.7 shows the effect of this cross-talk error on the single ion state readout fidelity of the FFNN alongside the overlap between the DPSF of the two ions. The DPSF overlap between the ions is significant for separations of less than 2 px, consistent with the considered Airy radius of 1.95 px, significantly reducing the fidelity of the state readout. For separations greater than 2 px, the overlap in the DPSF rapidly tapers out, as such, the single ion fidelity approaches an asymptote approximately at the single ion detection fidelity shown in Table 4.1.

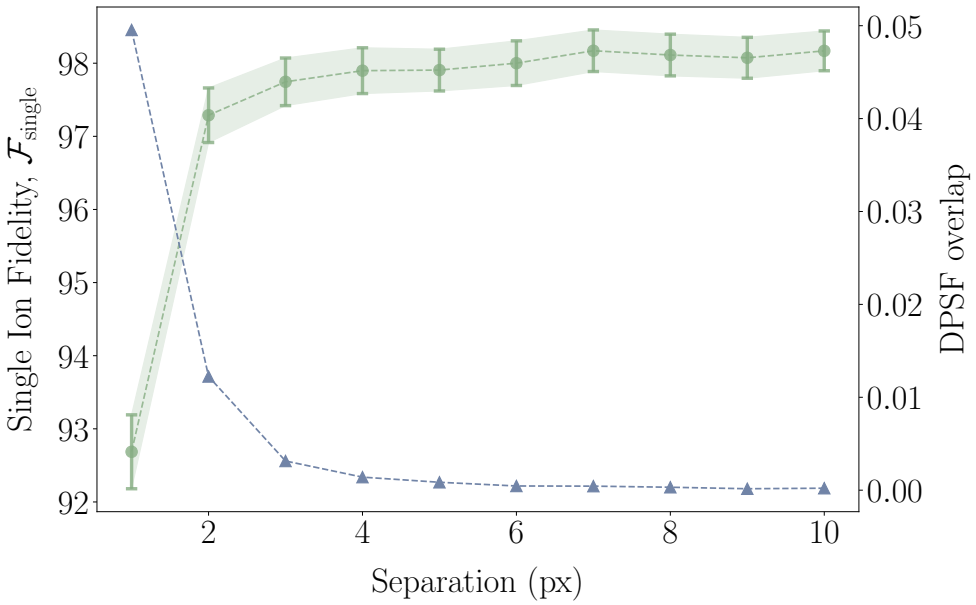


Figure 4.7: **Effect of cross-talk on state readout fidelity.** Single ion state readout fidelity (green circles) of the state readout FFNN for a $N = 2$ ion system with varying separation between the ions. The overlap between the DPSF (blue triangle) of the 2 ions. Notice, significant DPSF overlap between the 2 ions results in reduced single ion state readout fidelity. However, For > 2 px ionic separation, the DPSF overlap of the 2 ions become insignificant and the single ion fidelity converges.

We now move on to consider a system of more than two ions where the ions are in the chain configuration, the condition for this is shown in Section 1.1. We choose the trapping parameters such that the ions have a mean separation from their nearest neighbour of approximately 4 px on the detector. Observe that the single ion state readout fidelity of the FFNN is within margin of error regardless of the number of ions in Fig. 4.8. This is explained by the cross-talk the ions experience. As we have seen in Fig. 4.7, for a separation of 4 px or more, the error in the fidelity is negligible. Therefore, the single ion fidelity here is not limited by cross-talk.

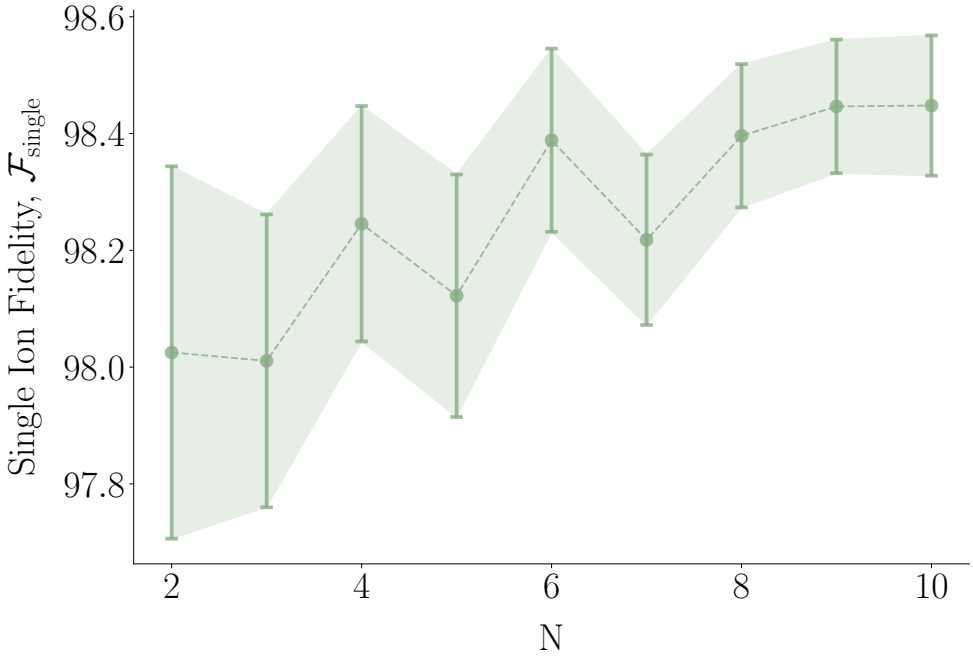


Figure 4.8: **Performance of neural network state readout for varying number of ions.** Single ion state readout fidelity of the FFNN for varying number of ions in a chain configuration. The mean separation between the ions are approximately 4 px. As expected, as the ionic separation is significant, the single ion state readout fidelity is approximately independent of the number of ions.

The key takeaways from the investigation of the use of the FFNN for state readout in a trapped-ion system is as follows:

- The neural network functions as a linear classifier greatly resembling traditional adaptive threshold methods.
- The neural network is capable of learning the full structure of the PSF (without error) or the region of interest (ROI) (with error) of the ions. This allows the neural network to locate the ion on the detector.

A key benefit of the neural network also limits its scalability, as the neural network learns the centroid of the PSF of each ion in the determination of the state of the trapped-ion system, the neural network is sensitive to any movement in the ions and changes in the

number of ions. Any such change would require the neural network be retrained for that specific system. However, with the knowledge of the inner workings of the FFNN, we imagine a scheme insensitive to such changes. We split the state readout process into two distinct stages,

1. Determine an ROI for each ion.
2. Distinguish the state of each ion using the corresponding ROI.

4.3 Two-stage state readout protocol

4.3.1 Region of interest detection

In this section, we develop a simple algorithm to solve the first stage of the proposed two stage state readout process, i.e., determine the ROI for each ion in the trapped-ion system. We take inspiration from object detection algorithms which use the convolutional neural network (CNN) architecture. Our problem, however, is much simpler. In object detection algorithms, the size of the object on the image and the key features of the object are unknown. The CNN has to be flexible enough to accommodate the object regardless of its size on the image and has to learn the features of the object through training. In contrast, we are aware of the size of the ROI, with the parameters considered in the previous section, the ROI for an ion is approximately 3 by 3 pixels. Furthermore, we know the characteristics of an ion shown on the detector, given by the DPSF of the ion. We envision the appropriate application of this information to an algorithm with CNN like structure would allow it to accurately determine the region of interest for the ions in the system.

Calibration Image

Locating all the ions in the system requires the ions to fluoresce regardless of their state. We achieve this by imaging the ions when performing Doppler cooling on the trapped-ion system [42]. Doppler cooling is performed before any QIP experiment as cooling is required to reduce the errors that occur during the experiments. During Doppler cooling, we choose a ground state and an excited state for the ion with an optical transition frequency. By applying a laser beam red-detuned from the transition, the ion preferentially scatters light depending on its movement relative to the cooling beam. The ion scatters more light when it moves towards the laser and scatters less light when it moves away from the laser. The net

result is the ion reduces its speed in the direction of the laser and thus energy. To ensure the ion stays in the cooling cycle, additional lasers are applied to depopulate the other excited states of the ion by driving them to the ground state. By collecting the photons scattered during Doppler cooling, we obtain an image of the ions where all ions are bright regardless of their initial state before Doppler cooling, we call this image the calibration image, X_{mn}^{calib}. When compared to the images used for state readout, the calibration image created during Doppler cooling can have significantly higher signal-to-noise ratio as we are able to use higher intensities for the cooling beam, compared to the detection beam, and collect the scattered photons for a longer period time, compared to the detection time. This calibration image is then used to determine the ROI for each ion in the system. The assumption we make is that after the cooling procedure, during the QIP experiment, the ions do not appreciably shift in position. This is a safe assumption as movements in the ions are controlled during QIP experiments as they could lead to significant errors. For a trapped-ion system of $N = 10$ ions and axial trap strength of $0.2 \times 2\pi$ MHz cooled such that the average phonon number in the axial modes of $\langle n \rangle \approx 1$, the uncertainty in the positions of the ions is ≈ 10 nm which corresponds to ≈ 0.01 px on the detector. The positional uncertainty of the ions increases as the average phonon number increases and decreases as the number of ions and trap strength increases. Achieving this level of cooling for the axial mode can be experimentally challenging for larger number of ions, e.g., the Doppler cooling limit of the $^{171}Yb^{+}$ ions trapped with the aforementioned trapping strength corresponds to an average phonon number, $\langle n \rangle \approx 50$. Other cooling techniques, such as, resolved sideband cooling, can be employed to further cool the axial modes.

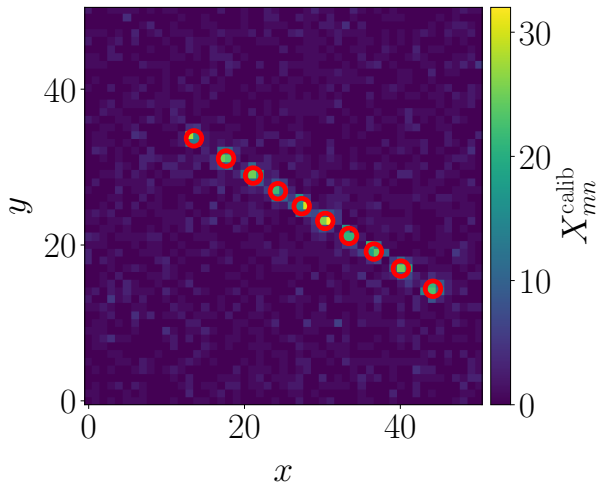


Figure 4.9: **Calibration image.** Example calibration image with 10 ions (red circles) in a linear configuration with a mean separation of 4 px. Here, $\lambda_0 = 127$ and $\lambda_{\text{noise}} = 0.6$.

The procedure for the region of interest (ROI) detection algorithm, example shown in Fig. 4.10, takes the following simple form:

1. Filter out the noise in the calibration image using a threshold method,

$$\tilde{X}_{mn}^{\text{calib}} = \begin{cases} 0 & \text{if } X_{mn}^{\text{calib}} < \theta_1 \\ X_{mn}^{\text{calib}} & \text{otherwise} \end{cases} \tag{4.16}$$

   Due to the high signal-to-noise ratio of the calibration image, a threshold method is capable of effectively filtering out the noise.

2. Convolve the filtered image with the 3 by 3 DPSF of an ion, while keeping the dimensions of the image. The resultant image tells us the overlap between the DPSF and the distribution of photons in each 3 by 3 region in the image.

3. Check every 3 by 3 pixel region of the resultant image in step 2 and determine if the center pixel has the maximum value in the region and is greater than a threshold, θ_2. If both of these conditions are true, we determine that an ion is at the center pixel of the 3 by 3 pixel region.

A caveat of this procedure is that the ions have to be sufficiently separated, as significant overlap in the DPSF of the ions would result in a rise of errors where multiple ions are detected as a single ion.

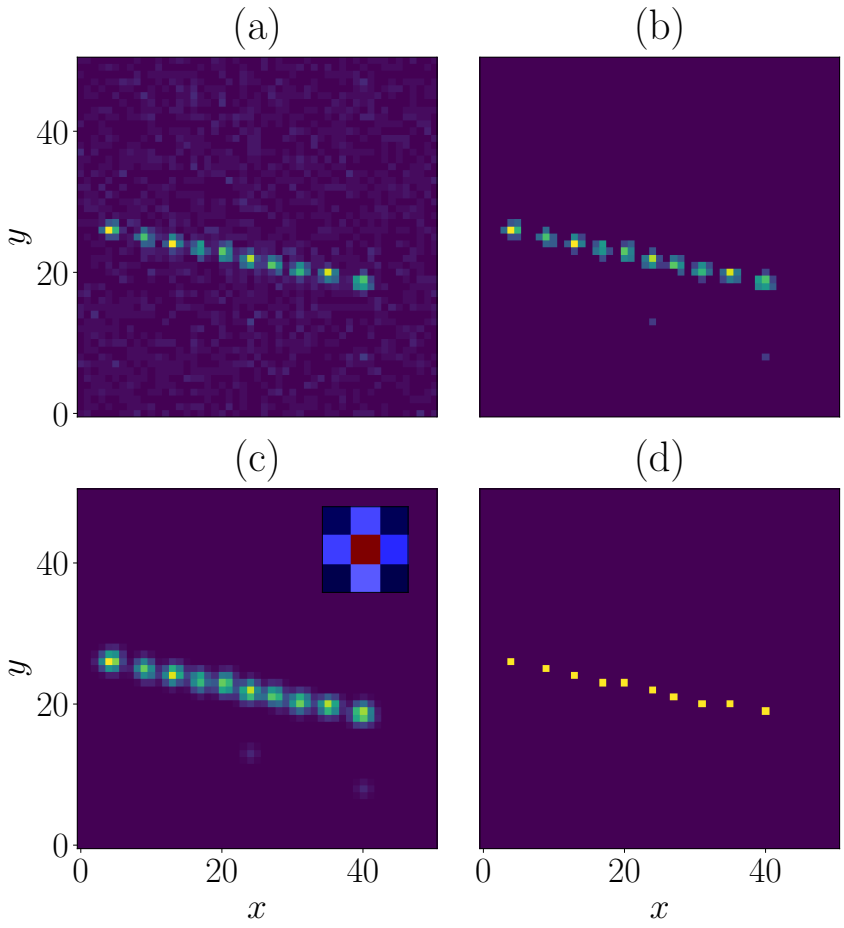


Figure 4.10: **ROI detection algorithm.** (a) Initial calibration image fed into the ROI detection algorithm, (b) calibration image after filtering out the noise with a threshold, $\theta_1 = 5$, (c) image after using the DPSF (inset) as the kernel for convolution on the filtered image and (d) predicted positions of the ions (yellow pixels), obtained using a threshold, $\theta_2 = 5$.

The ROI detection algorithm was evaluated with a dataset composed of 10000 calibration images with 10 ions each, the parameters for the thresholds were empirically determined, $\theta_1 = 5$ and $\theta_2 = 5$. We found that the algorithm accurately predicted the position of 81.0(4)% of the ions, completely missed 2.7(4)% of the ions and mischaracterized the position, i.e., the predicted position of the ions were off by a single pixel, either horizontally, vertically or diagonally, of 16.3(4)% of the ions in the dataset it was tested on. For

the cases where the algorithm mischaracterized the position, we further investigated the overlap between the predicted region of interest and the DPSF of the ion and significant overlap between the two were observed. This implies that the use of the ROIs from these mischaracterized positions should not have a significant effect on the fidelity of state readout that follows.

4.3.2 State readout

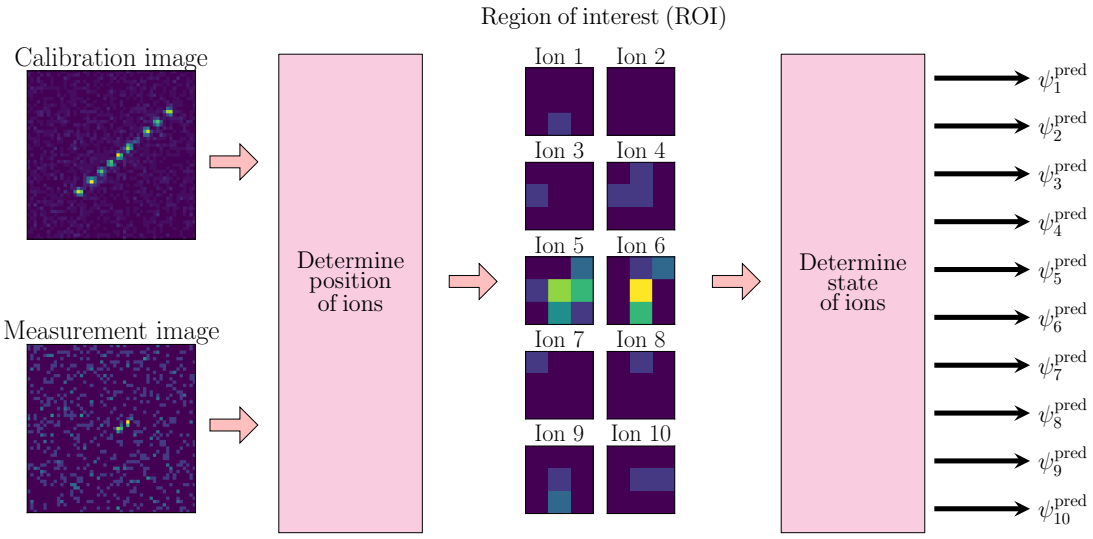


Figure 4.11: **Two-stage state readout algorithm.** (a) The calibration and measurement images are first fed into the ROI algorithm. (b) The ROI algorithm determines the ROIs for the ions and the measurement image is decomposed into the (c) individual measurement images for each ion. Finally, the (d) state readout FFNN is applied to the individual measurement images for each ion and outputs the predicted state for each individual ions, ψ_i^{pred}.

After the determination of the position of each ion, we apply the FFNN described in Section 4.2.1 on the ROI of the ions to perform the state readout on the measurement image. After training, we were able to achieve a single ion state readout fidelity of $\mathcal{F}_{\text{single}} = 0.9915(30)$ for the cases where the ROI detection algorithm was able to locate the ion. Notice the improvement in the fidelity as compared to Table 4.1, this is explained by the explicit filtering out of the pixels not relevant to the state readout of an ion. Consequently, the

neural network is less prone to overfitting to the training dataset here. A full description of the overall protocol is given in Fig. 4.11

4.3.3 Conclusion

In this chapter, we discussed the theory of measurement in a trapped-ion system. We studied the mechanism used by a feedforward neural network to perform state detection. The feedforward neural network demonstrated the ability to learn characteristics of the PSF of the ion. However, the feedforward neural network is not ideal due to its senstivity to the position of the ions. We conceptualized a two-stage process for state detection that would be capable of performing state readout by first determining a region of interest for the ions before performing the state readout. This eliminates the need to retrain the neural network for varying number of ions and any shifts in the position of the ions on the detector.

Chapter 5

Conclusion and outlook

In summary, we have presented several optimization problems that arise in the development of trapped-ion systems. In the programming of a trapped-ion system for the simulation of arbitrary spin models, we demonstrated that a simple feedforward neural network was capable of approximating the inverse function. This allowed the feedforward neural network to obtain accurate results for the control parameters in a relatively short amount of time, compared to traditional approaches, on a consumer grade GPU workstation.

Furthermore, we have introduced a hybrid ion trapping architecture that further enhances the utility of trapped-ion systems by enabling control over the phonon modes of the system. We developed protocols that would allow us to program these phonon modes for a wide range of applications. Additionally, we studied a promising application of the protocol to mitigate the disadvantages in multi-species trapped-ion systems.

For the readout of trapped-ion system, we developed a two-stage algorithm that would allow additional flexibility in quantum information processing experiments.

For the future, we envision studying further applications of the hybrid trapped-ion system proposed in this thesis. We also look to further refine the two-stage state readout algorithm and experimentally implement the procedure.

References

[1] Y. H. Teoh, M. Drygala, R. G. Melko, and R. Islam, Machine learning design of a trapped-ion quantum spin simulator, Quantum Science and Technology **5**, 024001 (2020).

[2] Y. H. Teoh, M. Sajjan, Z. Sun, F. Rajabi, and R. Islam, Manipulating phonons of a trapped-ion system using optical tweezers, Phys. Rev. A **104**, 022420 (2021).

[3] R. P. Feynman, Simulating physics with computers, International Journal of Theoretical Physics **21**, 467 (1982).

[4] J. I. Cirac and P. Zoller, Goals and opportunities in quantum simulation, Nature Physics **8**, 264 (2012).

[5] A. Aspuru-Guzik and P. Walther, Photonic quantum simulators, Nature Physics **8**, 285 (2012).

[6] I. Bloch, J. Dalibard, and S. Nascimbène, Quantum simulations with ultracold quantum gases, Nature Physics **8**, 267 (2012).

[7] R. Blatt and C. F. Roos, Quantum simulations with trapped ions, Nature Physics **8**, 277 (2012).

[8] Y. Wang, M. Um, J. Zhang, S. An, M. Lyu, J.-N. Zhang, L.-M. Duan, D. Yum, and K. Kim, Single-qubit quantum memory exceeding ten-minute coherence time, Nature Photonics **11**, 646 (2017).

[9] A. Friedenauer, H. Schmitz, J. T. Glueckert, D. Porras, and T. Schaetz, Simulating a quantum magnet with trapped ions, Nature Physics **4**, 757 (2008).

[10] K. Kim, M.-S. Chang, S. Korenblit, R. Islam, E. E. Edwards, J. K. Freericks, G.-D. Lin, L.-M. Duan, and C. Monroe, Quantum simulation of frustrated ising spins with trapped ions, Nature **465**, 590 (2010).

[11] J. T. Barreiro, M. Müller, P. Schindler, D. Nigg, T. Monz, M. Chwalla, M. Hennrich, C. F. Roos, P. Zoller, and R. Blatt, An open-system quantum simulator with trapped ions, Nature **470**, 486 (2011).

[12] J. W. Britton, B. C. Sawyer, A. C. Keith, C. C. J. Wang, J. K. Freericks, H. Uys, M. J. Biercuk, and J. J. Bollinger, Engineered two-dimensional ising interactions in a trapped-ion quantum simulator with hundreds of spins, Nature **484**, 489 (2012).

[13] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, Rev. Mod. Phys. **91**, 045002 (2019).

[14] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization (2014), arXiv:1412.6980 [cs.LG] .

[15] M. D. Zeiler, Adadelta: An adaptive learning rate method (2012), arXiv:1212.5701 [cs.LG] .

[16] T. Miki, M. Ito, Y. Hirata, Y. Kushitani, M. Shimada, and J. Shirakashi, Computational properties of ising spin model on spin connection parameters, in *2019 IEEE 19th International Conference on Nanotechnology (IEEE-NANO)* (2019) pp. 121–124.

[17] C. Muschik, M. Heyl, E. Martinez, T. Monz, P. Schindler, B. Vogell, M. Dalmonte, P. Hauke, R. Blatt, and P. Zoller, U(1) wilson lattice gauge theories in digital quantum simulators, New Journal of Physics **19**, 103020 (2017).

[18] S. Korenblit, D. Kafri, W. C. Campbell, R. Islam, E. E. Edwards, Z.-X. Gong, G.-D. Lin, L.-M. Duan, J. Kim, K. Kim, and C. Monroe, Quantum simulation of spin models on an arbitrary lattice with trapped ions, New Journal of Physics **14**, 095024 (2012).

[19] J. Adler and O. Öktem, Solving ill-posed inverse problems using iterative deep neural networks, Inverse Problems **33**, 124007 (2017).

[20] L.-F. Arsenault, R. Neuberg, L. A. Hannah, and A. J. Millis, Projected regression method for solving fredholm integral equations arising in the analytic continuation problem of quantum physics, Inverse Problems **33**, 115007 (2017).

[21] R. Fournier, L. Wang, O. V. Yazyev, and Q. Wu, Artificial neural network approach to the analytic continuation problem, Phys. Rev. Lett. **124**, 056401 (2020).

[22] A. Sørensen and K. Mølmer, Quantum computation with ions in thermal motion, Phys. Rev. Lett. **82**, 1971 (1999).

[23] K. Kim, M.-S. Chang, R. Islam, S. Korenblit, L.-M. Duan, and C. Monroe, Entanglement and tunable spin-spin couplings between trapped ions using multiple transverse modes, Phys. Rev. Lett. **103**, 120502 (2009).

[24] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, Automatic differentiation in PyTorch, in *NIPS Autodiff Workshop* (2017).

[25] T. Olsacher, L. Postler, P. Schindler, T. Monz, P. Zoller, and L. M. Sieberer, Scalable and parallel tweezer gates for quantum computing with long ion strings, PRX Quantum **1**, 020316 (2020).

[26] Y.-C. Shen and G.-D. Lin, Scalable quantum computing stabilised by optical tweezers on an ion crystal, New Journal of Physics **22**, 053032 (2020).

[27] X. Hou and W. Cheng, Optical tweezers, in *Encyclopedia of Biophysics* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2013) pp. 1800–1807.

[28] M. Endres, H. Bernien, A. Keesling, H. Levine, E. R. Anschuetz, A. Krajenbrink, C. Senko, V. Vuletic, M. Greiner, and M. D. Lukin, Atom-by-atom assembly of defect-free one-dimensional cold atom arrays, Science **354**, 1024 (2016).

[29] D. Barredo, S. de Léséleuc, V. Lienhard, T. Lahaye, and A. Browaeys, An atom-by-atom assembler of defect-free arbitrary two-dimensional atomic arrays, Science **354**, 1021 (2016).

[30] D. Barredo, V. Lienhard, S. de Léséleuc, T. Lahaye, and A. Browaeys, Synthetic three-dimensional atomic structures assembled atom by atom, Nature **561**, 79 (2018).

[31] A. Omran, H. Levine, A. Keesling, G. Semeghini, T. T. Wang, S. Ebadi, H. Bernien, A. S. Zibrov, H. Pichler, S. Choi, J. Cui, M. Rossignolo, P. Rembold, S. Montangero, T. Calarco, M. Endres, M. Greiner, V. Vuletić, and M. D. Lukin, Generation and manipulation of schrödinger cat states in rydberg atom arrays, Science **365**, 570 (2019).

[32] C. Schneider, M. Enderlein, T. Huber, and T. Schaetz, Optical trapping of an ion, Nature Photonics **4**, 772 (2010).

[33] A. Lambrecht, J. Schmidt, P. Weckesser, M. Debatin, L. Karpa, and T. Schaetz, Long lifetimes and effective isolation of ions in optical and electrostatic traps, Nature Photonics **11**, 704 (2017).

[34] J. Schmidt, A. Lambrecht, P. Weckesser, M. Debatin, L. Karpa, and T. Schaetz, Optical trapping of ion coulomb crystals, Phys. Rev. X **8**, 021028 (2018).

[35] R. W. Gerchberg, A practical algorithm for the determination of phase from image and diffraction plane pictures, Optik **35**, 237 (1972).

[36] K. G. Johnson, J. D. Wong-Campos, A. Restelli, K. A. Landsman, B. Neyenhuis, J. Mizrahi, and C. Monroe, Active stabilization of ion trap radiofrequency potentials, Review of Scientific Instruments **87**, 053110 (2016).

[37] I. V. Inlek, C. Crocker, M. Lichtman, K. Sosnova, and C. Monroe, Multispecies trapped-ion node for quantum networking, Phys. Rev. Lett. **118**, 250502 (2017).

[38] K. Sosnova, A. Carter, and C. Monroe, Character of motional modes for entanglement and sympathetic cooling of mixed-species trapped-ion chains, Phys. Rev. A **103**, 012610 (2021).

[39] A. Seif, K. A. Landsman, N. M. Linke, C. Figgatt, C. Monroe, and M. Hafezi, Machine learning assisted readout of trapped-ion qubits, Journal of Physics B: Atomic, Molecular and Optical Physics **51**, 174006 (2018).

[40] M. Acton, K.-A. Brickman, P. C. Haljan, P. J. Lee, L. Deslauriers, and C. Monroe, Near-perfect simultaneous measurement of a qubit register, Quantum Info. Comput. **6**, 465–482 (2006).

[41] J. D. Wong-Campos, K. G. Johnson, B. Neyenhuis, J. Mizrahi, and C. Monroe, High-resolution adaptive imaging of a single atom, Nature Photonics **10**, 606 (2016).

[42] Olmschenk, Steven Matthew, *Quantum Teleportation Between Distant Matter Qubits*, Ph.D. thesis, University of Michigan (2009).

APPENDICES

Appendix A

Derivation of motional eigenvectors for the trapped-ion system

The Lagrangian of the trapped ion system has the form:

$$\mathcal{L} = \sum_i \frac{M_i \dot{\mathbf{r}}_i^2}{2} - \phi(\{\mathbf{r_i}\}_i). \tag{A.1}$$

Here, $\dot{\mathbf{r}}_i$ is the velocity of ion-i and ϕ is the potential of the system.
Let,

$$\mathbf{q} = (x_1, x_2, ..., x_N, y_1, ..., y_N, z_1, ..., z_N), \tag{A.2}$$

$$\bar{M} = (M_1, M_2, ..., M_N, M_1, ..., M_N, M_1, ..., M_N). \tag{A.3}$$

The equation of motion for the above Lagrangian can obtained from the Euler-Lagrange equation:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}\right) = \frac{\partial \mathcal{L}}{\partial \mathbf{q}}, \tag{A.4}$$

$$\bar{M}_k \ddot{\mathrm{q}}_k = \frac{\partial \phi}{\partial \mathrm{q}_k}. \tag{A.5}$$

As we are considering small perturbations about the equilibrium position of the ions, $\mathbf{q}^*$, we linearize the above equation to obtain:

$$\bar{M}_k \ddot{\mathrm{q}}_k = \left.\frac{\partial \phi}{\partial \mathrm{q}_k}\right|_{\mathbf{q}^*} + \sum_j \frac{\partial^2 \phi}{\partial \mathrm{q}_k \partial \mathrm{q}_j}(\mathrm{q}_j - \mathrm{q}_j^*). \tag{A.6}$$

Consider the following change of coordinates, from $\mathbf{q}$ to $(\mathbf{q} - \mathbf{q}^*)$. Physically, this corresponds to measuring displacements between the ions and the equilibrium position instead of the absolute position of the ion. Mathematically, without loss of generality, we are able to set $\mathbf{q}^* = 0$.

As the equilibrium position corresponds to the position where the ions do not experience a force, the first term on the right hand side of the equation goes to zero, we obtain:

$$\bar{M}_k \ddot{\mathrm{q}}_k = \sum_j \frac{\partial^2 \phi}{\partial \mathrm{q}_k \partial \mathrm{q}_j} \mathrm{q}_j = \sum_j [\mathrm{Hess}(\phi)]_{kj} \mathrm{q}_j. \tag{A.7}$$

Here, $\mathrm{Hess}(\,\cdot\,)$ is the hessian with respect to all $3N$ coordinates.
Divide both sides by the factor, $\sqrt{\bar{M}_k}$ we get:

$$\ddot{\mathrm{q}}_k \sqrt{\bar{M}_k} = \sum_j \frac{1}{\sqrt{\bar{M}_k}} [\mathrm{Hess}(\phi)]_{kj} \frac{\sqrt{\bar{M}_j}}{\sqrt{\bar{M}_j}} \mathrm{q}_j, \tag{A.8}$$

$$= \sum_j \mathrm{A}_{kj} \mathrm{q}_j \sqrt{\bar{M}_j}. \tag{A.9}$$

Assume $\mathrm{q}_j \sqrt{\bar{M}_j} = \mathrm{b}_j$, where $\mathbf{b}$ is an eigenvector of $\mathbf{A}$ with eigenvalue λ, we obtain:

$$\ddot{\mathrm{q}}_k \sqrt{\bar{M}_k} = \lambda \mathrm{q}_k \sqrt{\bar{M}_k}. \tag{A.10}$$

This is the harmonic oscillator equation. This system would oscillate at frequency $\sqrt{\lambda}$.

Note that we mentioned that we assume $\mathrm{q}_j \sqrt{\bar{M}_j} = \mathrm{b}_j$, however, $\mathrm{q}_j \sqrt{\bar{M}_j}$ is a mass-weighted coordinate of the ion. To obtain the eigenvectors of the system only in terms of the motion of the ion, this mass-weighting has to be undone.

$$\mathrm{q}_j = \bar{\mathrm{b}}_j = \frac{\mathrm{b}_j}{\sqrt{\bar{M}_j}}. \tag{A.11}$$

After applying the transform, we renormalize the motional eigenvector $\bar{\mathbf{b}}$.

In the case where $M_j = M$ for all j we obtain:

$$\mathrm{q}_j = \bar{\mathrm{b}}_j = \mathrm{b}_j. \tag{A.12}$$

This explains why for a single species system the eigenvectors of the $\mathbf{A}$-matrix directly corresponds to the motional participation of the ion in the phonon mode, whereas, for a multi-species system, a transform on the eigenvectors (Eq. 1.8) is required.

Appendix B

Derivation of properties of the phonon mode control problems

Non-existence of solutions for the control of eigenfrequencies

An important property in this derivation is that the **A**-matrix is a real symmetric matrix with a unitary eigenvector matrix.
The A matrix is related to its eigenfrequencies according to the following equation:

$$\mathrm{A}_{ij} = \sum_m \mathrm{B}_{im} \left(\mathrm{w}_m\right)^2 \mathrm{B}_{jm}. \tag{B.1}$$

Consider taking the Frobenius norm on both sides, and note that the Frobenius norm of a matrix is invariant under a unitary transform.

$$\|\mathbf{A}\|_2 = \|\mathbf{w} \circ \mathbf{w}\|_2 \tag{B.2}$$

Consider the minimum possible value of the Frobenius norm for a feasible **A**-matrix, this corresponds to setting the diagonal elements to zero. As the off-diagonal elements of a feasible **A**-matrix of a system of more than $N = 1$ ion is non-zero, we have a lower bound for the eigenfrequencies of the system, mathematically,

$$\|\mathbf{w} \circ \mathbf{w}\|_2 \geq \sum_{i \neq j} (\mathrm{A}_{ij}^{\mathrm{conv}})^2 \tag{B.3}$$

This proves we are incapable of obtaining an arbitrary set of eigenvectors using optical tweezers if the conventional trapped-ion system is determined.

Non-uniqueness of solutions for the control of eigenfrequencies

Due to the symmetry in the equilibrium positions of the ions, it is trivial to deduce that reflecting the optical tweezer strengths about the center of the trapped-ion chain results in identical phonon mode eigenfrequencies.

Non-existence of solutions for the control of eigenvectors

To prove this we consider a trivial system of a diagonal matrix. Diagonal matrices only have eigenvectors corresponding to the identity matrix. As such we suspect that even in the non-trivial case, there are eigenvectors unachievable by control over the diagonal elements of the matrix.

Non-uniqueness of solutions for the control of eigenvectors

Note that the identity matrix is a diagonal matrix and maps any vector onto itself. Therefore, the addition of a multiple of the identity matrix does not change the eigenvectors of any matrix. As the the identity matrix is diagonal, adding a multiple of the identity to a feasible $\mathbf{A}$-matrix does not violate the constraint of feasibility. This implies that when a solution exist for the control of the eigenvectors, there are an infinite number of solutions that are related by multiples of the identity.